**СОДЕРЖАНИЕ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

УО «ВГТУ» ПП.008 1-40 05 01-01 ПЗ

Разраб.

Шершень С. Н.

Провер.

Реценз.

Н. Контр.

Утверд.

Разработка информационной системы учёта программного обеспечения

Лит.

Листов

1

УО «ВГТУ» ИСАП гр. ИТС-6

[Введение 3](#_Toc73289936)

[1 Анализ объекта 5](#_Toc73289937)

[1.1 Описание предметной области 5](#_Toc73289938)

[1.2 Построение концептуальной модели предметной области 5](#_Toc73289939)

[2 Постановка задачи 8](#_Toc73289940)

[2.1 Определение требований к программной системе. 8](#_Toc73289941)

[2.2 Описание аналогов системы 9](#_Toc73289942)

[2.3 Обзор и обоснование выбора инструментальных средств 13](#_Toc73289943)

[3 Проектирование 17](#_Toc73289944)

[3.1 Разработка архитектуры программного продукта 17](#_Toc73289945)

[3.2 Проектирование структур хранения данных 19](#_Toc73289946)

[3.3 Описание реализации вариантов использования 23](#_Toc73289947)

4 [Реализация 25](#_Toc73289948)

[4.1 Разработка классов информационной системы 25](#_Toc73289949)

[4.1 Разработка интерфейса программного продукта 27](#_Toc73289950)

[4.2 Разработка алгоритмов реализации вариантов использования 33](#_Toc73289951)

[4.3 Модульное тестирование алгоритмов реализации вариантов использования 34](#_Toc73289952)

[5 Системное тестирование 37](#_Toc73289953)

[5.1 Функциональное тестирование 38](#_Toc73289954)

[5.2 Оценка безопасности 39](#_Toc73289955)

[5.3 Тестирование производительности 41](#_Toc73289956)

[6 Экономическая часть 43](#_Toc73289957)

[6.1 Расчет трудоемкости разработки программного обеспечения 43](#_Toc73289958)

[6.2 Определение стоимости разработки программного обеспечения 43](#_Toc73289959)

[6.3 Расчёт экономического эффекта от внедрения программного продукта 53](#_Toc73289960)

[7 Охрана труда 56](#_Toc73289961)

[8 Промышленная экология 70](#_Toc73289962)

[9 Ресурсосбережение 74](#_Toc73289963)

[Заключение 78](#_Toc73289964)

[Список использованной литературы 79](#_Toc73289965)

# Введение

Процесс автоматизации сопровождает общество с момента его зарождения, она внутренне присуща его развитию. В методологии он определяется как замещение процессов человеческой деятельности процессами технических устройств. Однако процесс развития предполагает, что сначала человек должен познать процесс и только потом он сможет переложить реализацию на машины. Невозможно автоматизировать ни один процесс, пока не будут получены знания о том, как, а главное, зачем его осуществляет человек [1].

Автоматизация производственных процессов особенно необходима для того, чтобы упорядочить деятельность организации. Безусловно, некоторую структуру можно создать с помощью служебных инструкций и обучения персонала, однако в период активного развития инструкции пришлось бы переписывать, а частая смена персонала свела бы на нет все усилия. Поэтому возникает необходимость в автоматизации производственных процессов, которые смогут развиваться наряду с прогрессом.

Современные предприятия используют большое количество компьютерной техники и программного обеспечения для организации работы персонала. Рано или поздно каждая компания приходит к тому, что требуется навести порядок в инфраструктуре предприятия, получить контроль над размещением техники, учету компьютеров и программного обеспечения, установленного на них. Для упрощения ведения контроля возникает необходимость в создании автоматизированной системы.

Целью дипломной работы является разработка информационной системы учета программного обеспечения

Система позволит автоматизировать процесс хранения и поиска различной информации о компьютерах, их атрибутах, таких как мощность процессора, видеокарте, жесткому диску; о программном обеспечении, учитывая его тип и производителя, уменьшит количество ошибок при вводе данных, исключит возможность потери данных при длительном хранении, увеличит производительность сотрудников благодаря снижению времени на поиск и отображение информации.

Для достижения поставленной цели необходимо решить следующие взаимосвязанные задачи:

* провести анализ предметной области и сформировать требования к программному продукту;
* провести обзор существующих систем для решения поставленной задачи;
* спроектировать информационную систему «Сотрудники»;

# 1 Анализ объекта

## 1.1 Описание предметной области

Разрабатываемая информационная система предназначена для автоматизации учёта программного обеспечения.

В данной системе организована работа с различными наиболее часто используемыми данными о программном обеспечении и используемых компьютерах. Благодаря этому, данная система может быть использована в различных как по профилю деятельности, так и по масштабу организациях.

При начале использования нового компьютера в базу заносится общая информация о компьютере, такая как его название, местоположение, используемые видеокарта, жесткий диск и мощность процессора, а также размеры оперативной памяти.

В течение времени, каждому компьютеру устанавливается то или иное программное обеспечение (информация о котором хранится в отдельной таблице в базе данных).

Факт установки программного обеспечения на компьютер фиксируется запись даты установки.

## 1.2 Построение концептуальной модели предметной области

При построении концептуальной модели предметной области была сформирована функциональная структура программного средства [2]. Она состоит из следующих подсистем:

Подсистема для работы с компьютерами. Предназначена для добавления, удаления и редактирование информации о всех компьютерах. Также показывает список всего программного обеспечения выбранного компьютера и позволяет добавлять, изменять и удалять информации об установке программного обеспечения на выбранный компьютер.

Подсистема для работы с программным обеспечением. Предназначена для добавления, удаления и редактирование информации о всем программном обеспечении. Также показывает список всех компьютеров, на которых установлено выбранное программное обеспечение и позволяет добавлять, изменять и удалять информации об установке этого программного обеспечения на выбранный компьютер.

Подсистема справочник программного обеспечения. Предназначена для добавления, удаления и редактирования информации о такой информации ПО как производитель и тип. Также показывает ПО выбранных производителя или типа и позволяет взаимодействовать с ним.

Подсистема справочник компьютеров. Предназначена для добавления, удаления и редактирования информации о такой информации компьютера как мощность процессора, видеокарте и жесткому диску. Также показывает все компьютеры, на которых установлены данные средства и позволяет взаимодействовать с ними.

Функциональная структура представлена на рисунке 1.1.



Рисунок 1.1 – Функциональная структура

При построении концептуальной модели выделены следующие объекты:

* Main – Главная форма приложения с меню доступа
* Compute – для работы с компьютерами
* ComputerGuide – для работы со справочником компьютера
* Software – для работы с программным обеспечением
* SoftwareGuide – для работы со справочником программного обеспечения
* ReportByPC – для отображения отчета по компьютерам
* ReportBySoftware – для отображения отчета по программному обеспечению

# 2 Постановка задачи

## 2.1 Определение требований к программной системе.

Основные функции представлены в виде UML диаграммы вариантов использования.

При разработке диаграммы вариантов использования проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования. Актером называется любая сущность, взаимодействующая с системой извне [4]. Вариант использования служит для описания сервисов, которые система предоставляет актеру. Диаграмма вариантов использования может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов.

Диаграмма вариантов использования представлена на рисунке 2.1.

В разрабатываемой системе возможна работа одного типа пользователя, соответственно на диаграмме вариантов использования представлен один актер:

Администратор обладает всеми функциями приложения, что позволяет ему взаимодействовать с данными всех таблиц приложения.

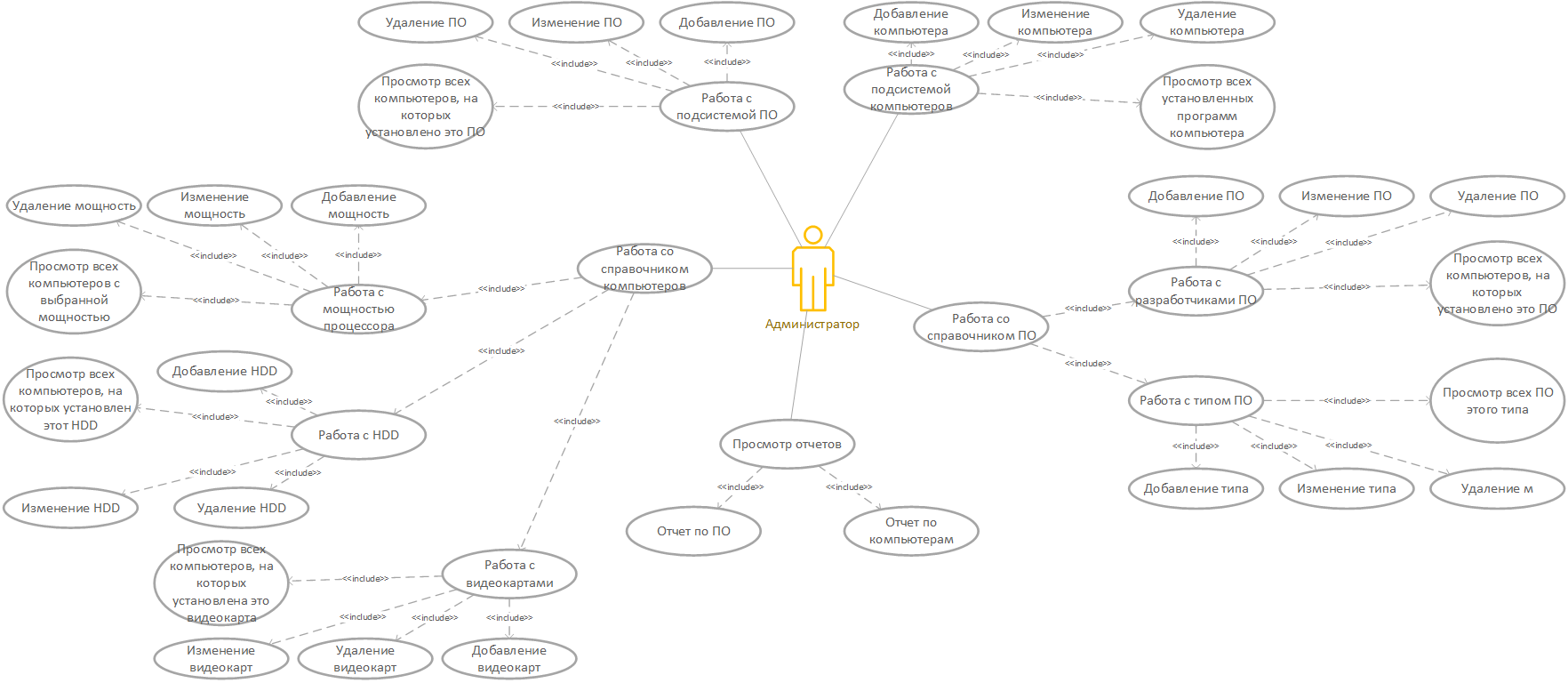


Рисунок 2.1 – Диаграмма вариантов использования

Для полноценной работы системы компьютер пользователя должен удовлетворять следующим аппаратным требованиям:

- процессор AMD Athlon X4 840 3.10 GHz;

- ОЗУ: не менее 4Гб;

- HDD: не менее 250Гб;

- видеоадаптер: DirectX9;

- видеопамять: не менее 512 МБ;

Кроме того, необходимы следующие программные компоненты:

- Windows 7 и выше;

- MS SQL Server 2018 – для машины-сервера;

- MS .NET Framework 4.6.1;

## 2.2 Описание аналогов системы

**IBM License Metrics Tool (ILMT)**

IBM License Metrics Tool (ILMT) — инструмент, позволяющий контролировать расход лицензий с метриками PVU (Processor Value Units), RVU (Resource Value Units) и некоторых других типов лицензий. Ограничением этого инструмента является невозможность измерять расход пользовательских типов лицензий, таких как AU (Authorized User) и CC (Concurrent User). Возможно, это ограничение будет устранено в последующих версиях ILMT. Инструмент доступен для скачивания на сайте производителя, но только клиентами IBM [5].

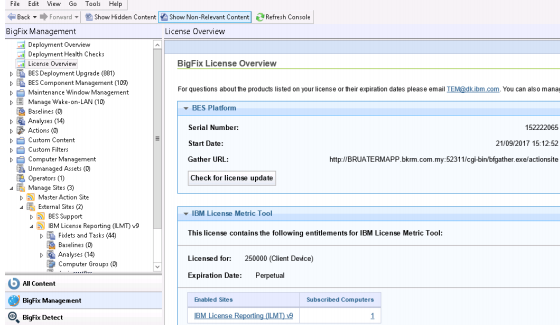


Рисунок 2.2 - IBM License Metrics Tool (ILMT)

Организации, заключившие соглашение IBM Passport Advantage, как правило, применяют для расчета необходимого количества лицензионных прав метод Неполных мощностей (Subcapacity). Этот метод позволяет оптимизировать затраты на лицензии для сложных вычислительных сред. При применении метода Неполных мощностей использование инструмента ILMT становится обязательным условием лицензионного соглашения с IBM.

Для работы с ILMT необходимо также установить другой программный продукт — IBM BigFix. ILMT работает с данными, которые собираются программой IBM BigFix. При использовании IBM BigFix только для поддержки работы ILMT, лицензии на IBM BigFix приобретать не требуется. Для сбора данных об установленном в организации программном обеспечении IBM используются агенты BigFix, устанавливаемые на каждое устройство, с которого должна собираться информация. Данные о структуре виртуализации собираются IBM BigFix при помощи дополнительного коннектора VM Manager Tool.

ILMT не позволяет учитывать количество приобретенных прав на программные продукты, но позволяет хранить перечень приобретенных лицензий. Соответственно, в самой программе нельзя выполнить сравнение имеющихся и используемых лицензионных прав.

Для учета лицензионных прав и вычисления соответствия имеющихся и используемых прав, необходимо применять дополнительное программное обеспечение с возможностями управления программными активами (SAM).

**MAP (Microsoft) Toolkit**

MAP (Microsoft) Toolkit — набор инструментов для анализа и оценки ИТ-инфраструктуры организации, работающий без использования программ-агентов. За формирование отчетов, используемых для управления лицензиями на ПО, в MAP Toolkit отвечает утилита Software Usage Tracker [6].

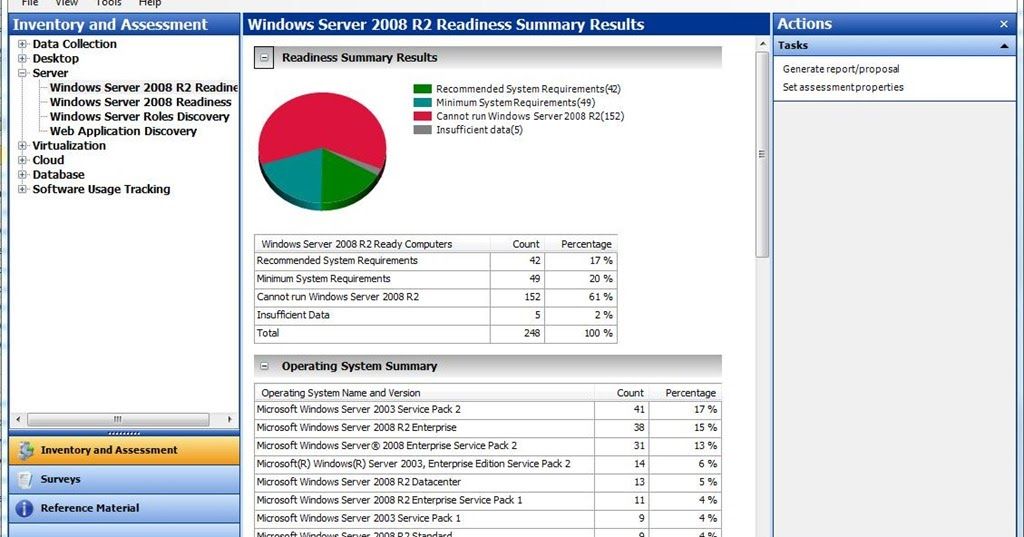


Рисунок 2.3 - MAP (Microsoft) Toolkit

Утилита формирует отчеты по следующим программным продуктам Microsoft: Windows Server, SharePoint Server, Exchange Server, SQL Server и System Center Configuration Manager. По данным продуктам осуществляется контроль лицензий со следующими метриками: Per-server, Per-processor, User CAL, Device CAL. Соответственно, для указанных программных продуктов вы можете вычислить и оценить свои потребности в серверных, процессорных и клиентских (CAL) лицензиях.

MAP не умеет учитывать имеющиеся в организации лицензионные права и осуществлять их сравнение с используемым правами, для чего также необходимо использовать дополнительное программное обеспечение.

**Usage Hub (HPE)**

Usage Hub (HPE) – специальный инструмент, созданный Hewlett Packard Enterprise для контроля лицензий и позиционируемый как ”free service”. Usage Hub умеет собирать и просматривать информацию об имеющихся лицензиях на программы HPE, сроке их действия и объеме использования этих лицензий [7].

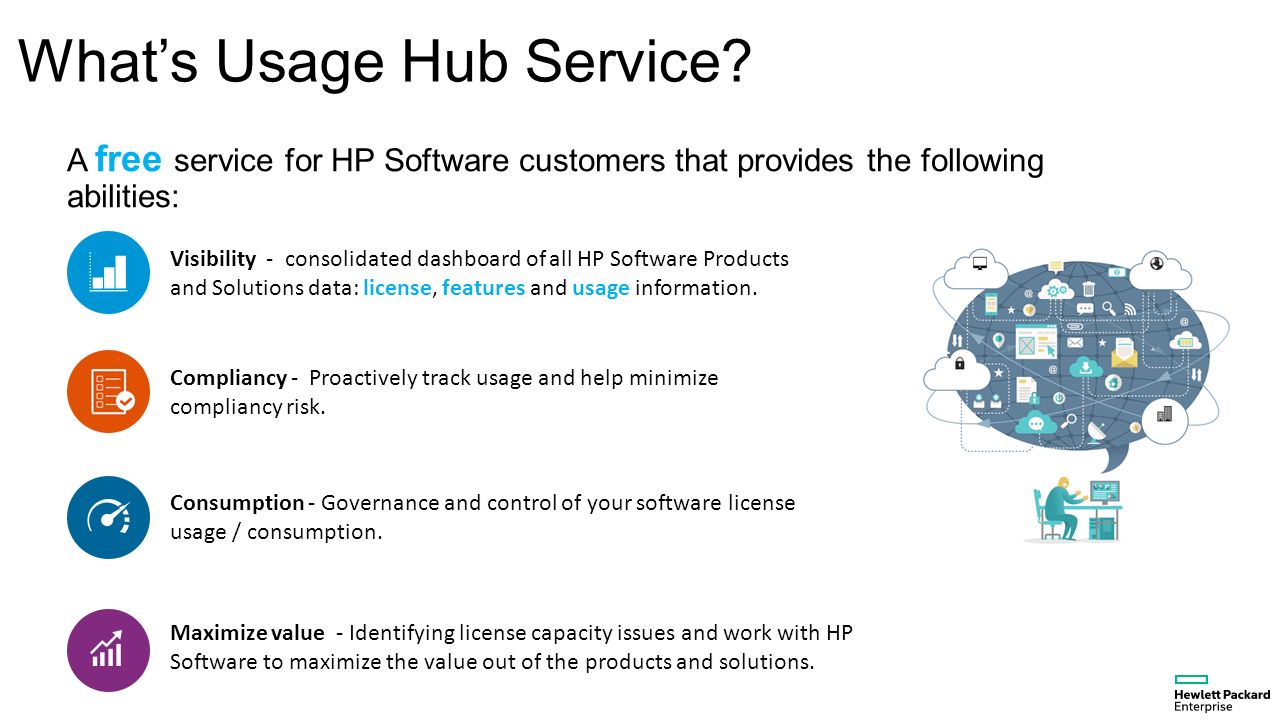


Рисунок 2.4 - Usage Hub (HPE)

Часть информации Usage Hub собирает с помощью программ-агентов, а часть безагентским способом. Для получения полной информации о лицензиях и их сроке действия, необходимо дополнительно настраивать взаимодействие Usage HUB с HPE AutoPass License Server. И сам Usage Hub, и его агенты могут быть установлены в операционных средах Windows и Linux.

Неоспоримое преимущество этого решения – возможность получения отчетов о лицензионном соответствии (compliance), формируемым как на текущий момент, так и за определенный исторический период времени, благодаря автоматическому сбору информации об установленных лицензиях и сроке их действия. Впрочем, здесь стоит сделать оговорку: Usage Hub не предназначен для выполнения аудита лицензионного соответствия в организации, о чем прямо указано в документации к продукту.

## 2.3 Обзор и обоснование выбора инструментальных средств

Разработка базы данных для хранения всей необходимой информации осуществлялось по средствам СУБД MS SQLServer.

Microsoft SQL Server — система управления реляционными базами данных, разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка [8].

MS SQL Server – это платформа для решения критически важных задач в масштабе предприятия, обладающая высокой доступностью, повышенной производительностью и безопасностью. Решение представляет собой хорошо масштабируемый, полностью реляционный, быстродействующий сервер, способный обрабатывать большие объемы данных для клиент-серверных приложений.

Среди основных преимуществ MS SQL Server 2012 можно выделить следующие:

- шифрование как всей базы данных, так и всех файлов;

- минимизируется нагрузка на сеть и обеспечивается оптимальная производительность, благодаря сжатию исходящего потока;

- управление производится на основе политик, что дает возможность удобно управлять не только одним, но и несколькими экземплярами СУБД;

- разработчикам баз данных доступно большое количество различных функций.

Рекордная производительность MS SQL Server обеспечивается новыми технологиями работы с памятью, что помогает предприятиям ускорить свой бизнес и реализовать новые сценарии работы. Кроме того, SQL Server позволяет использовать новые гибридные облачные решения и пользоваться новыми преимуществами облачных вычислений.

Расширенные функции безопасности, в сочетании со встроенными, удобными для использования инструментами и управляемым доступом к данным, позволяют организации выполнить требования строгих политик соответствия нормам.

Сервер поддерживает широкий спектр средств разработки и достаточно прост при взаимодействии с различными средствами разработки клиентских приложений. SQL Server обрабатывает запросы от пользователей и отправляет пользователю результаты запроса. Таким образом улучшается время отклика.

Для разработки клиентского приложения была выбрана среда MicrosoftVisualStudio 2019. VisualStudio – это средство для разработки программного обеспечения, которое имеет большой функционал и возможности для отладки и развертывания приложений на различных платформах. В комплект разработчика входят интегрированная среда разработки (IDE) и инструменты тестирования для всех этапов процесса разработки программного обеспечения. Интегрированная среда разработки представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения [9].

MicrosoftVisualStudio поддерживает широкий спектр популярных языков программирования. Это хороший инструмент для профессиональных разработчиков программного обеспечения, которые хотят создавать коммерческие программы.

Основные преимущества MicrosoftVisualStudio:

- поддержка множества языков при разработке.

- редактор VisualStudio оптимизирует работу программиста по кодированию. В ходе того, как вводится новый код, на экране будут появляться подсказки. Для ускорения выполнения многих задач в VisualStudio предоставлено большое количество клавиатурных комбинаций. Существует набор средств быстрой переработки, которые позволяют быстро усовершенствовать структуру кода, не отрываясь от процесса программирования.

- интуитивный стиль кодирования. По умолчанию VisualStudio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев. Такие незначительные отличия делают код более удобным для чтения и менее подверженным ошибкам.

- более высокая скорость разработки. Многие из функциональных возможностей VisualStudio направлены на то, чтобы помогать разработчику делать свою работу как можно быстрее. Удобные функции, вроде функции IntelliSense, функции поиска и замены и функции автоматического добавления и удаления комментариев позволяют разработчику работать быстро и эффективно.

- возможности отладки. Предлагаемые в VisualStudio инструменты отладки являются наилучшим средством для отслеживания ошибок и диагностирования странного поведения. Разработчик может выполнять свой код по строке за раз, устанавливать интеллектуальные точки прерывания, при желании сохраняя их для использования в будущем, и в любое время просматривать текущую информацию из памяти.

Для разработки приложения был выбран язык C#. C# предназначен для разработки приложений, которые выполняются в среде .NET Framework. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C. C# — это объектно-ориентированный язык, который поддерживает все правила объектно-ориентированного программирования: инкапсуляцию, наследование и полиморфизм. Язык поддерживает перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML [10].

C# имеет множество функций, обеспечивающих надежность и устойчивость приложений. Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и их восстановлению. Типобезопасная структура языка делает невозможным чтение из неинициализированных переменных, индексацию массивов за пределами их границ или выполнение непроверенных приведений типов.

# 3 Проектирование

## 3.1 Разработка архитектуры программного продукта

Приложение будет построено на базе архитектуры MVC.

Термин модель-представление-контроллер (model-view-controller) используется с конца 70-х гг. прошлого столетия. Этот шаблон происходит из проекта Smalltalk, выполнявшегося в Xerox PARC, где он был задуман как способ организации ранних приложений с графическим пользовательским интерфейсом. Некоторые нюансы первоначального шаблона MVC были связаны с концепциями, специфичными для Smalltalk, такими как экраны и инструменты, но более широкие понятия по-прежнему применимы к приложениям - и особенно хорошо они подходят для веб-приложений [11].

Взаимодействие с приложением MVC осуществляется в соответствии с естественным циклом действий пользователя и обновлений представления, при котором предполагается, что представление не содержит информации о состоянии. Это прекрасно сочетается с запросами и ответами HTTP, которые лежат в основе веб-приложения.

Более того, инфраструктура MVC принудительно применяет разделение ответственности - модель предметной области и логика контроллера отделены от пользовательского интерфейса. В веб-приложении это означает, что HTML-разметка хранится отдельно от остальной части приложения, что упрощает и облегчает сопровождение и тестирование.

Появление платформы Ruby on Rails привело к возобновлению всеобщего интереса к MVC, и он остается шаблоном реализации архитектуры MVC. С тех пор появилось много других инфраструктур MVC, и все они демонстрировали преимущества архитектуры MVC - разумеется, это относится и к ASP.NET MVC.

Если оперировать высокоуровневыми понятиями, то архитектурный шаблон MVC означает, что приложение MVC будет разделено, по крайней мере, на три части, которые описаны ниже:

Модели, содержащие или представляющие данные, с которыми работают пользователи. Они могут быть простыми моделями представлений, которые только представляют данные, передаваемые между представлениями и контроллерами, или же они могут быть моделями предметной области, которые содержат бизнес-данные, а также операции, преобразования и правила для манипулирования этими данными.

Представления, применяемые для визуализации некоторой части модели в виде пользовательского интерфейса.

Контроллеры, которые обрабатывают поступающие запросы, выполняют операции с моделью и выбирают представления для визуализации пользователю.

Модели - это определение "вселенной", в которой функционирует ваше приложение. Например, в банковском приложении модель представляет все аспекты банковской деятельности, поддерживаемые приложением, такие как расчетные счета, главная бухгалтерская книга и кредитные лимиты для клиентов, равно как и операции, которые могут использоваться для манипулирования данными в модели, такие как внесение денежных средств и списание со счетов. Модель отвечает также за сохранение общего состояния и целостности данных - например, удостоверяясь, что все транзакции внесены в главную книгу, а клиент не снимает со счета больше денежных средств, чем имеет на то право, или больше, чем имеется в распоряжении самого банка.

Модели определяются также и тем, за что они не отвечают: модели не имеют дела с визуализацией пользовательских интерфейсов или обработкой запросов, т.к. это ответственность представлений и контроллеров.

Представления содержат логику, необходимую для отображения элементов модели пользователю - и ничего более. Они не имеют никаких прямых сведений о модели и не обмениваются данными непосредственно с ней. Контроллеры являются шлюзом между представлениями и моделью - запросы поступают от клиента и обслуживаются контроллером, который выбирает подходящее представление для отображения пользователю и, если требуется, соответствующую операцию, которая должна быть выполнена над моделью.

Каждая часть архитектуры MVC является четко определенной и самодостаточной - это то, что называется разделением ответственности. Логика, которая манипулирует данными в модели, содержится только в модели, логика, отображающая данные - только в представлении, а код, который обрабатывает запросы и ввод пользователей - только в контроллере. При четком разграничении всех частей приложение будет легче сопровождать и расширять в течение его срока существования, независимо от того, насколько большим оно станет.

## 3.2 Проектирование структур хранения данных

В процессе разработки системы были выделены следующие сущности, используемые для хранения данных:

* Разработчик (Код, Название)
* Тип ПО (Код, Тип)
* Программное обеспечение (Код, Название, Версия, Разработчик, Тип, Приблизительный размер (МБ))
* Процессор (Код, Процессор)
* Видеокарта (Код, Видеокарта)
* HDD (Код, HDD)
* Компьютер (Код, Название, ИП адрес, Местоположение, Процессор, Оперативная память, Видеокарта, Объем виде опамяти, HDD, Объем HDD)
* Установленное ПО (Код, ПК, ПО, Дата установки)

Взаимодействие между таблицами организовано при помощи ключей. Схема данных имеет вид (рис. 3.1):

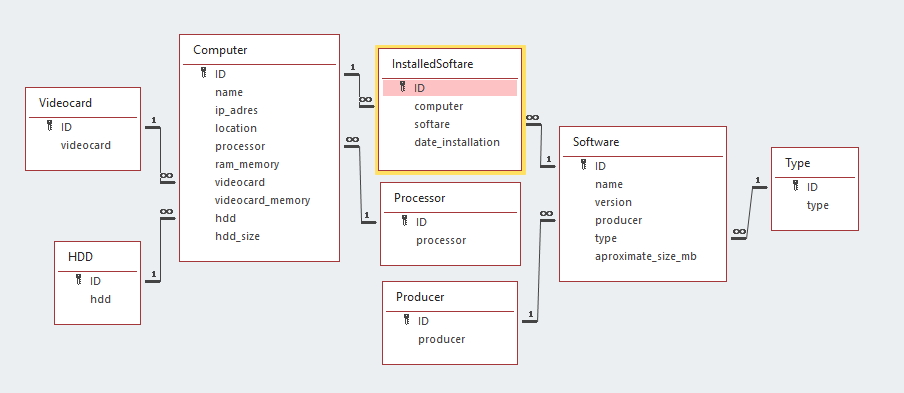


Рисунок 3.1 –Схема данных

Для создания базы данных по указанным сущностям был создан следующий код создания базы данных.

Листинг 3.1 – Код создания базы данных (MS SQL Server)

DROP DATABASE "SoftareAccounting";

CREATE DATABASE "SoftareAccounting";

GO

USE "SoftareAccounting";

GO

CREATE TABLE "Producer" (

"ID" INTEGER NOT NULL IDENTITY(1,1),

"producer" VARCHAR(255),

PRIMARY KEY ("ID")

)

GO

CREATE TABLE "Type" (

"ID" INTEGER NOT NULL IDENTITY(1,1),

"type" VARCHAR(255),

PRIMARY KEY ("ID")

)

GO

CREATE TABLE "Software" (

"ID" INTEGER NOT NULL IDENTITY(1,1),

"name" VARCHAR(255),

"version" VARCHAR(255),

"producer" INTEGER,

"type" INTEGER,

"aproximate\_size\_mb" VARCHAR(255),

PRIMARY KEY ("ID"),

FOREIGN KEY ("producer") REFERENCES "Producer"("ID"),

FOREIGN KEY ("type") REFERENCES "Type"("ID")

)

GO

CREATE TABLE "HDD" (

"ID" INTEGER NOT NULL IDENTITY(1,1),

"hdd" VARCHAR(255),

PRIMARY KEY ("ID")

)

GO

CREATE TABLE "Processor" (

"ID" INTEGER NOT NULL IDENTITY(1,1),

"processor" VARCHAR(255),

PRIMARY KEY ("ID")

)

GO

CREATE TABLE "Videocard" (

"ID" INTEGER NOT NULL IDENTITY(1,1),

"videocard" VARCHAR(255),

PRIMARY KEY ("ID")

)

GO

CREATE TABLE "Computer" (

"ID" INTEGER NOT NULL IDENTITY(1,1),

"name" VARCHAR(255),

"ip\_adres" VARCHAR(255),

"location" VARCHAR(255),

"processor" INTEGER,

"ram\_memory" VARCHAR(255),

"videocard" INTEGER,

"videocard\_memory" VARCHAR(255),

"hdd" INTEGER,

"hdd\_size" VARCHAR(255),

PRIMARY KEY ("ID"),

FOREIGN KEY ("processor") REFERENCES "Processor"("ID"),

FOREIGN KEY ("videocard") REFERENCES "Videocard"("ID"),

FOREIGN KEY ("hdd") REFERENCES "HDD"("ID")

)

GO

CREATE TABLE "InstalledSoftare" (

"ID" INTEGER NOT NULL IDENTITY(1,1),

"computer" INTEGER,

"softare" INTEGER,

"date\_installation" DATETIME,

PRIMARY KEY ("ID"),

FOREIGN KEY ("computer") REFERENCES "Computer"("ID"),

FOREIGN KEY ("softare") REFERENCES "Software"("ID")

)

GO

В реляционной базе данных каждому объекту и сущности реального мира соответствуют кортежи отношений. И любое отношение должно обладать первичным ключом. Ключ – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности [12].

Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Каждое отношение должно обладать хотя бы одним ключом.

## 3.3 Описание реализации вариантов использования

Моделирование необходимо для понимания системы. Обычно, при этом единственной модели никогда не бывает достаточно. Наоборот, для понимания практически любой нетривиальной системы приходится разрабатывать большое количество взаимосвязанных моделей. В применении к программным системам это означает, что необходим язык, с помощью которого можно с различных точек зрения описать представления архитектуры системы на протяжении цикла ее разработки [13].

Диаграмма вариантов использования показывает, какая функциональность должна быть реализована в системе, основные функции, которые должны быть включены в систему, их окружение и взаимодействие функций с окружением.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых вариантов использования. При этом актером или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

В результате анализа предметной области и постановки задачи были определены варианты использования системы. Для реализации данных вариантов необходимо определить взаимодействие частей проектируемого программного средства. Диаграмма вариантов использования была отображена на рисунке 2.1.

# 4 Реализация

## 4.1 Разработка классов информационной системы

Windows Forms — это технология пользовательского интерфейса для .NET, представляющая собой набор управляемых библиотек, которые упрощают выполнение стандартных задач, таких как чтение из файловой системы и запись в нее. С помощью среды разработки, такой как Visual Studio, можно создавать интеллектуальные клиентские приложения Windows Forms, которые отображают информацию, запрашивают ввод пользователя и взаимодействуют с удаленными компьютерами по сети [14].

В Windows Forms форма — это визуальная поверхность, на которой выводится информация для пользователя. Обычно приложение Windows Forms строится путем добавления элементов управления в формы и создания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие. Приложение реагирует на эти события, как задано в коде, и обрабатывает события при их возникновении.

В Windows Forms предусмотрено множество элементов управления, которые можно добавлять в формы. Например, элементы управления могут отображать текстовые поля, кнопки, раскрывающиеся списки, переключатели и даже веб-страницы.

Используя функцию перетаскивания конструктора Windows Forms в Visual Studio, можно легко создавать приложения Windows Forms. Просто выделите элемент управления с помощью курсора и поместите его на нужное место в форме. Для преодоления трудностей, связанных с выравниванием элементов управления, конструктор предоставляет такие средства, как линии сетки и линии привязки. С помощью элементов управления FlowLayoutPanel, TableLayoutPanel и SplitContainer можно гораздо быстрее создавать сложные макеты форм.

Наконец, если нужно создать свои собственные элементы пользовательского интерфейса, пространство имен System.Drawing содержит широкий набор классов, необходимых для отрисовки линий, кругов и других фигур непосредственно на форме.

При создании формы, создается класс, соответствующий этой форме. Наше приложение сделано однооконным, т.е. будет только один класс отображения.

Тем не менее, для отображения и работы с данными в базе данных каждой таблице создаются свои классы адаптеры.

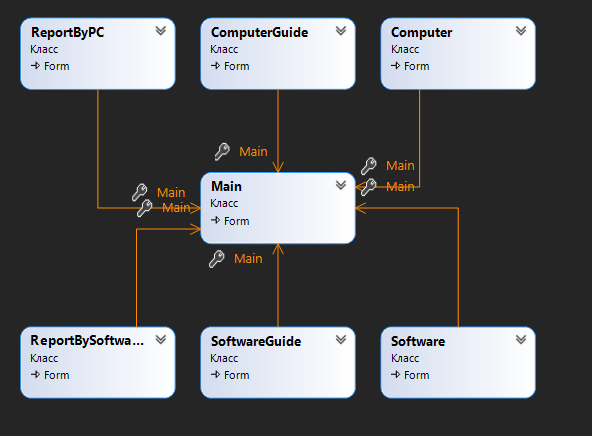


Рисунок 4.1 – Диаграмма классов

## [Разработка интерфейса программного продукта](#_Toc474749001)

Реализацию вариантов использования, а также описание взаимодействия системы с пользователем можно представить в виде скриншотов отображения смены форм посредством работы пользователя.

Как было описано в диаграмме вариантов использования, пользователь на начальном этапе имеет доступ к подсистемам работы с компьютерами, программами, справочнику для программ и справочнику по компьютерам, а также возможность просмотра отчетов.

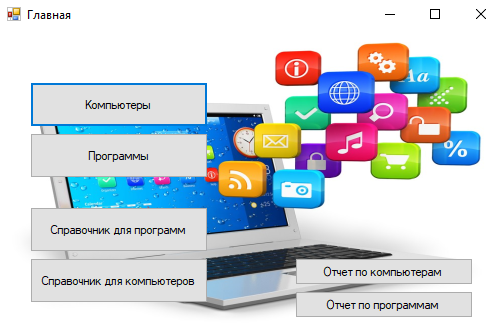


Рисунок 4.2 – Главная форма

При выборе пункта перехода к подсистеме компьютеров, пользователю будет отображен список всех компьютеров, под выбранным из которых будет отображен список всех установленных программ. Благодаря навигаторам над таблицами, пользователь может редактировать данные как в таблице компьютеров, так и связанные данные с выбранным компьютером по установке программного обеспечения.

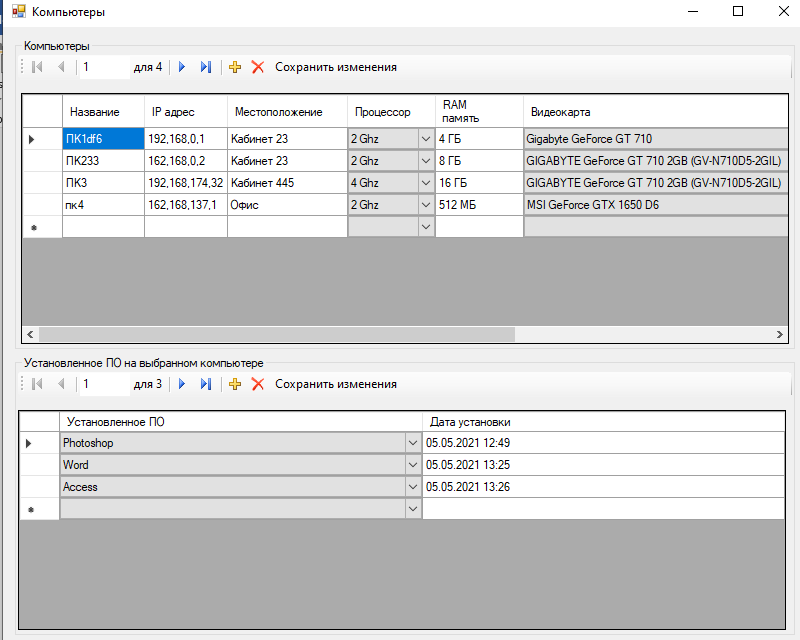


Рисунок 4.3 – Модуль работы с компьютерами

Модуль работы с программами представляет собой обратную функциональность к модулю работы с компьютерами.

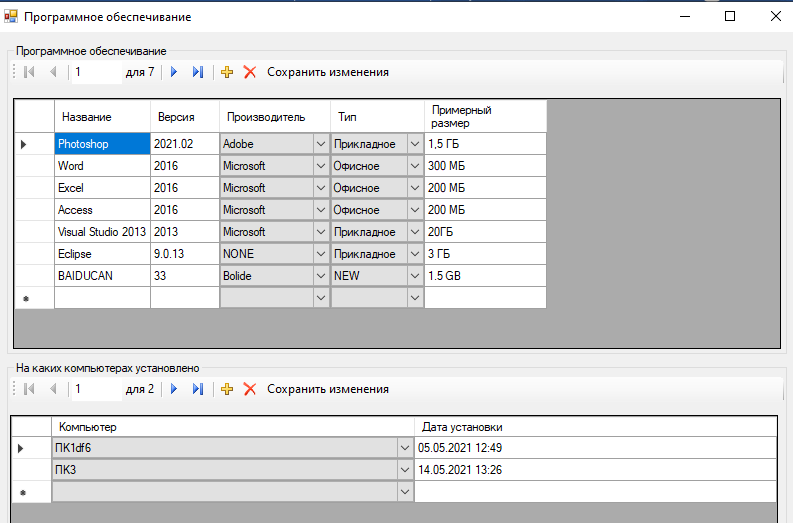


Рисунок 4.4 – Модуль работы с программным обеспечением

При выборе для работы модуля справочника для компьютера, будет отображена форма со вкладками, в каждой из которых расположена информация о компонентах компьютера, таких как мощность процессора, видеокарте и жестком диску.

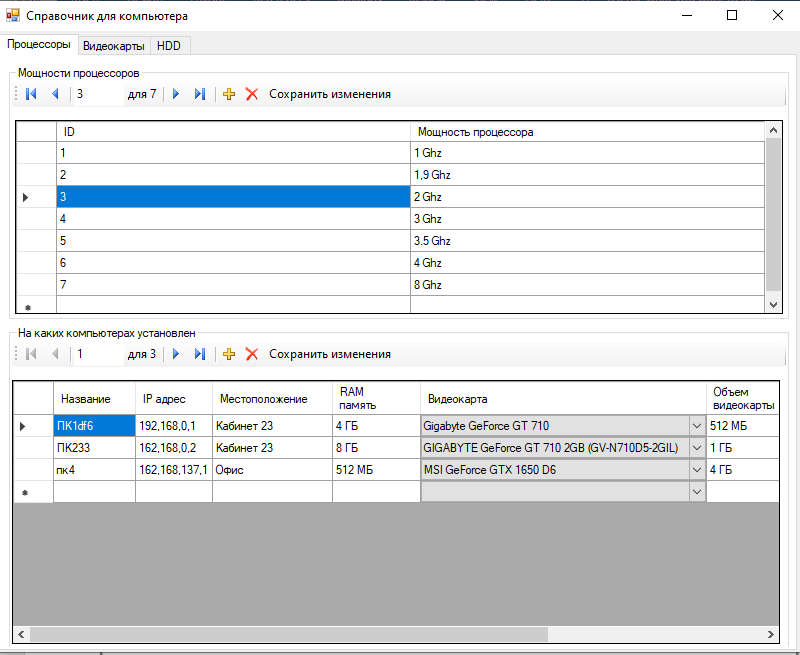


Рисунок 4.5 – Модуль справочника работы с компьютерами

Функциональность модуля справочника программного обеспечения похожа на предыдущую, и представляет такие же функции для работы с информацией, только в этот раз для атрибутов программного обеспечения, такой как авторы ПО и его тип.

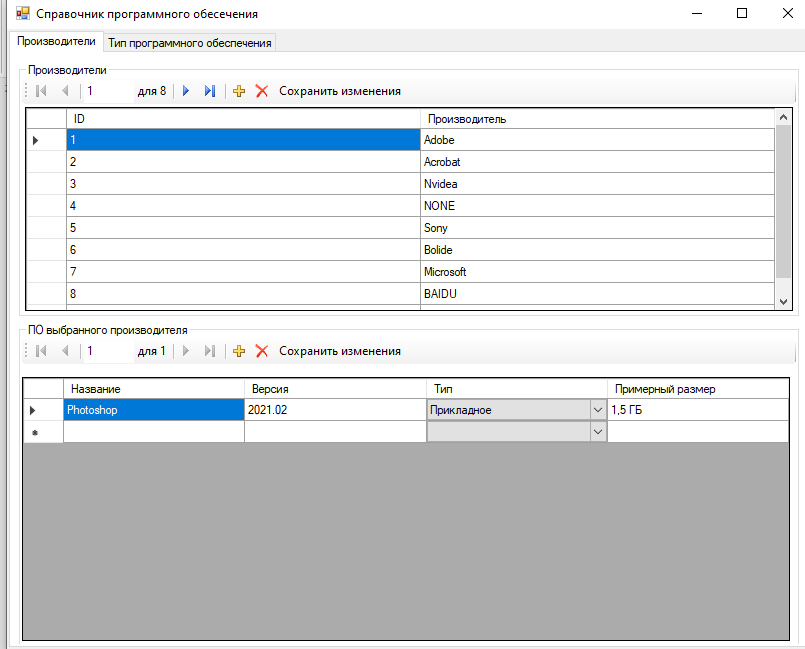


Рисунок 4.6 – Модуль справочника работы с программным обеспечением

Модуль вывода отчета по компьютерам выдает отчет всего программного обеспечения, установленного на каждом компьютере.

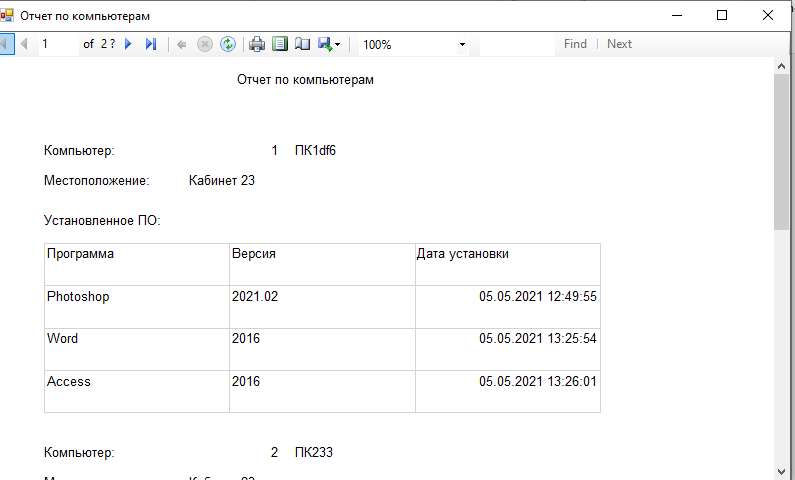


Рисунок 4.7 – Вывод отчета по компьютерам

Модуль же вывода отчета по программному обеспечению представляет собой вывод информации о программах и списках компьютеров, на которых установлено это программного обеспечение.

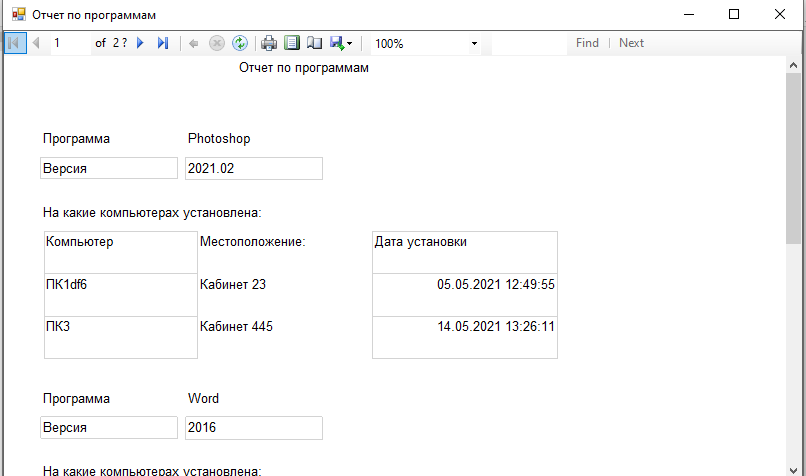


Рисунок 4.8 – Вывод отчета по программному обеспечению

## [Разработка алгоритмов реализации вариантов использования](#_Toc474749002)

При решении общей задачи, возникали более мелкие, для решения которых необходимо было использовать и разрабатывать алгоритмы. Несколько таких примеров будет показано ниже.

1. Сохранение информации в базу данных посредством BindingNavigator

Использование встроенного компонента помогает избегать множество проблем, когда работа идет только с одной таблицей. Такой компонент реализует все стандартные функции для отображения и изменения информации, однако не реализует метода сохранения информации. Так, для сохранения данных в таблицах, использующих данные только своей таблицы, используется следующий код.

private void toolStripButton1\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Все изменения в системе будут сохранены в базе данных.\r\nПродолжить сохранение?",

"Подтвердите сохранение",

MessageBoxButtons.YesNo) == DialogResult.Yes)

try

{

computerTableAdapter.Update(softareAccountingDataSet.Computer);

MessageBox.Show("Изменения сохранены");

loadData();

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

1. Создание адаптеров таблиц для генерации отчета

Для генерации отчета необходимо создать SQL запрос в базу данных. Однако, просто создать запрос недостаточно, поскольку необходимо также вывести результат в таблицу. Для вывода необходимо определить новый адаптер таблицы, добавив в него как зависимость все поля будущей таблицы.

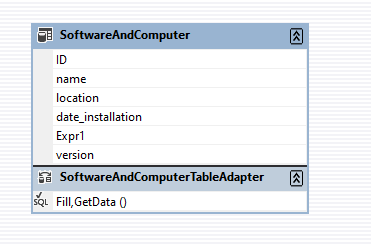


Рисунок 4.9 – Созданный адаптер таблицы для отчета

После создание адаптера, в метод его заполнение добавляется sql запрос на вывод всех требуемых полей таблицы, который будут участвовать в отчете.

SELECT Computer.ID, Computer.name, Computer.location, InstalledSoftare.date\_installation, Software.name AS Expr1, Software.version

FROM Computer INNER JOIN

InstalledSoftare ON Computer.ID = InstalledSoftare.computer INNER JOIN

Software ON InstalledSoftare.softare = Software.ID

## 4.3 Модульное тестирование алгоритмов реализации вариантов использования

Модульное тестирование, или юнит-тестирование - это процесс в программировании, который направлен на проверку отдельных небольших частей приложения, также называемых атомарными, которые можно исследовать изолированно от других подобных частей [15]. При выполнении данного тестирования могут проверяться как отдельные функции или методы классов, так и сами классы, взаимодействие классов, небольшие библиотеки и отдельные части приложения. Довольно часто данный вид тестирования реализуется с использованием специальных технологий и инструментальных средств автоматизации тестирования, значительно упрощающих и ускоряющих разработку соответствующих тест-кейсов. Таким образом, оценивая каждый элемент отдельно и подтверждая правильность его работы, установить проблему значительно проще чем, если бы элемент был частью системы.

Примеры тестирования алгоритмов реализации вариантов использования:

1 Тестирование вывода служащих при выборе подразделения (таблица 4.1).

Таблица 4.1 - Тестирование вывода списка установленного ПО при выборе компьютера

|  |  |  |  |
| --- | --- | --- | --- |
| **Краткое описание** | **Предварительные условия** | **Шаги для воспроизведения** | **Ожидаемый результат** |
| Тестирование вывода списка установленного ПО | Запустить приложение | Перейти на форму компьютеров.  Выбрать любой компьютер из списка | После выбора компьютера, в таблице ниже отображается список всего установленного ПО |

2 Тестирование добавления установленного ПО на ПК (таблица 4.2).

Таблица 4.2 - Тестирование добавления служащего

|  |  |  |  |
| --- | --- | --- | --- |
| **Краткое описание** | **Предварительные условия** | **Шаги для воспроизведения** | **Ожидаемый результат** |
| Тестирование добавления установленного ПО | Запустить приложение | Перейти на форму компьютеров.  Выбрать любой компьютер из списка  В список установленных по добавить новое ПО.  Нажать сохранить | При нажатии кнопки сохранить, пользователю будет показано окно подтверждения. После подтверждения добавленные данные отображаются в списке |

3. Тестирование изменение данных (таблица 4.3).

Таблица 4.3 - Тестирование изменения производителя

|  |  |  |  |
| --- | --- | --- | --- |
| Краткое описание | Предварительные условия | Шаги для воспроизведения | Ожидаемый результат |
| Тестирование изменения производителя | Запустить приложение | Перейти на форму справочника компьютеров.  Выбрать производителя из списка.  Изменить его данные.  Сохранить изменения | При нажатии кнопки сохранить, пользователю будет показано окно подтверждения. После подтверждения измененные данные отображаются в списке |

# 5 Системное тестирование

Системное тестирование — это тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям [16]. Системное тестирование относится к методам тестирования чёрного ящика, и, тем самым, не требует знаний о внутреннем устройстве системы. Тестирование бывает:

* функциональное тестирование;
* тестирование пользовательского интерфейса;
* тестирование совместимости;
* тестирование безопасности;
* тестирование производительности;
* автоматическое тестирование;
* интеграционное тестирование.

Испытания системы имеют две основные цели:

* показать разработчику и клиенту, что программное обеспечение отвечает заявленным требованиям. С точки зрения клиента это означает, что для каждой функциональности, желаемой с его стороны и записанной в документе требований, проведен как минимум в один тест (как правило, конечно, больше). В случае общедоступного программного обеспечения то, что в программном обеспечении протестированы все заданные основные свойства. Соответствующий данной цели тест называют вариацией (проверкой достоверности). Успешное вариация указывает, что система работает как надо;
* найти ситуации, когда программное обеспечение ведет себя ошибочно, нежелательно или не соответствует спецификации. Следовательно, поиск ошибок в этом смысле предназначен для того, чтобы ликвидировать нежелательное поведение системы, как, например, крах системы, нежелательное взаимодействие с другими системами, неправильные расчеты, поврежденные данные. Выполняющее эту задачу тестирование именуется тестированием дефектов (defect testing). Здесь являющийся успешным тест показывает действие ошибки системы, или, другими словами, находит в системе ошибку (к дальнейшему исправлению которой, и приступают).

## 5.1 Функциональное тестирование

Функциональное тестирование — это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определённых условиях решать задачи, нужные пользователям [17]. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает.

Функциональные требования включают в себя:

* функциональная пригодность;
* точность;
* способность к взаимодействию;
* соответствие стандартам и правилам;
* защищённость.

Тестирование и отладка программы являются наиважнейшими этапами разработки любых программных продуктов. Цель этого этапа - проверка правильности и точности реализации функций, выполнение которых возлагается на данный программный продукт. В случае выявления некоторых неточностей и ошибок необходимо проведение работ по их исправлению и доработке программного продукта до требуемого уровня.

На основе функций, которые должны быть протестированы в разработанной информационной обучающей системе, составлен чек-лист (checklist), приведенный в таблице 5.1.

Таблица 5.1 - Чек-лист

|  |  |  |  |
| --- | --- | --- | --- |
| **Тестируемый модуль** | | **Тестируемая функция** | **Результат** |
| Вход в систему | | Вход в систему | Выполнено успешно |
| Выход из системы | | Выход из системы | Выполнено успешно |
| Просмотр компьютеров | | Отображение информации и зависимых списков | Выполнено успешно |
| Просмотр информации выбранного компьютера | Отображение информации других таблиц на основе выбранного значения | | Выполнено успешно |
| Работа с производителями | Добавление, редактирование и удаление данных с выводом информации подчиненной таблицы | | Выполнено успешно |
| Работа с дисками | Добавление, редактирование и удаление данных с выводом информации подчиненной таблицы | | Выполнено успешно |
| Работа с видеокартами | Добавление, редактирование и удаление данных с выводом информации подчиненной таблицы | | Выполнено успешно |
| Работа с типом программного обеспечения | Добавление, редактирование и удаление данных с выводом информации подчиненной таблицы | | Выполнено успешно |

В качестве тестирования программного продукта был выбран тест-кейс (**Test Case**).

**Test Case** - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Тестирование производилось на ОС «Windows 10».

## 5.2 Оценка безопасности

Обеспечение безопасности информационных систем представляет собой ряд мероприятий, направленных на предотвращение несанкционированных воздействий на защищаемую информацию, а также её утечки.

Windows Forms содержит модель безопасности, основанную на коде (уровни безопасности задаются для кода, независимо от пользователя, запустившего код). Это дополнение к схемам безопасности, которые могут быть уже установлены на компьютере. К ним могут относиться данные в браузере (например, безопасность на основе зоны, доступная в Internet Explorer) или операционная система (например, безопасность на основе учетных данных Windows NT) [18].

До выпуска .NET Framework весь код, выполняемый на компьютере пользователя, имел те же права или разрешения на доступ к ресурсам, которые имел пользователь компьютера. Например, если пользователю был разрешен доступ к файловой системе, то код также получал доступ к файловой системе Если пользователю был разрешен доступ к базе данных, то и коду предоставлялся доступ к ней. Наличие таких прав или разрешений может быть допустимо для кода в исполняемых файлах, которые пользователь собственноручно установил на локальном компьютере, но оно неприемлемо в тех случаях, когда речь идет о потенциально вредоносном коде, поступившем из Интернета или локальной интрасети. Такой код не должен получать доступ к ресурсам компьютера пользователя без его разрешения.

В .NET Framework введена инфраструктура, называемая системой управления доступом для кода, которая позволяет отличать разрешения (или права) от прав, которые имеет пользователь. По умолчанию код, поступающий из Интернета или интрасети, может выполняться только в режиме частичного доверия. В режиме частичного доверия приложение подвергается ряду ограничений: помимо всего прочего, приложению отказано в доступе к локальным жестким дискам, и оно не может запускать неуправляемый код. .NET Framework управляет ресурсами, к которым коду разрешен доступ в зависимости от идентификатора этого кода: от того, откуда он поступил, имеет ли он сборки со строгими именами, подписаны ли они сертификатом и т. д.

Технология ClickOnce, используемая для развертывания Windows Forms приложений, помогает упростить разработку приложений, работающих в режиме частичного доверия, с полным доверием или с частичным доверием с повышенными разрешениями. Технология ClickOnce предоставляет такие функции, как повышение уровня разрешений и развертывание доверенных приложений, поэтому приложение может запрашивать полное доверие или повысить уровень разрешений от локального пользователя в ответственном способе.

## 5.3 Тестирование производительности

Тестирование производительности **— это** тестирование, которое проводится для определения скорости работы системы или её части при заданной нагрузке [19]. Тестирование производительности стремится учесть производительность уже на стадии проектирования и моделирования и системы, до начала основной стадии разработки.

Тестирование производительности служит таким типичным целям:

* для демонстрации того, что система удовлетворяет критериям производительности;
* для определения производительности, какой из двух или нескольких систем лучше;
* для определения, какой элемент нагрузки или часть системы приводит к снижению производительности.

Приложение запускается менее чем за 10 секунд, используя для своих нужд 30МБ памяти. На пике загрузки использует 7% ЦП.

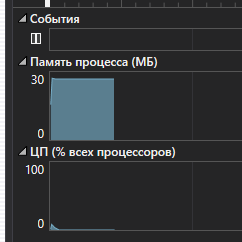


Рисунок 5.1 – Тестирование

Отображение данных не добавляет количество используемой памяти. При добавление новой записи в таблицу используемая память увеличилась до 38 МБ в первый раз, после чего осталось на том же уровне.

# 6 Экономическая часть

## 6.1 Расчет трудоемкости разработки программного обеспечения

Экономическая целесообразность разработки и внедрения программного обеспечения определяется экономическим эффектом, который будет получен произво­дителями при их реализации и потребителями при их использовании. По величине ожидаемого экономического эффекта принимается решение о целесообразности ин­вестиций в разработку того или иного программного продукта [20]. По характеру объекта вложений инвестиции в разработку программного обеспечения относят к интеллектуальным инвестициям.

При создании программного продукта важно оценить его себестоимости (затраты на разработку).

## 6.2 Определение стоимости разработки программного обеспечения

Затраты времени на разработку ПО могут также определяться эмпирическим путем. В этом случае затраты времени могут включать:

- затраты труда на подготовку и описание задачи – *tоп*;

- затраты труда на исследование алгоритма решения задачи – *tис*;

- затраты труда на разработку алгоритма (блок-схем) – tал;

- затраты труда на программирование алгоритма по блок-схеме – *tпр*;

- затраты труда на отладку программы – *tотл*;

- затраты труда на подготовку документов по задаче состоят из затрат труда на подготовку рукописей и времени на оформление документов – *tд.*

Суммарные затраты труда рассчитываются как сумма составных затрат труда по формуле (6.1):

. (6.1)

Расчет суммарных затрат времени представлен в таблице 6.1.

Таблица 6.1 - Ориентировочное распределение затрат времени

|  |  |  |
| --- | --- | --- |
| Вид работ | Трудоемкость в часах | |
| Всего, человеко-часов | в том числе машинное время, машино-часов |
| Подготовку и описание задачи | 14 | - |
| Исследование алгоритма решения задачи | 30 | - |
| Разработка алгоритма | 32 | - |
| Программирование алгоритма | 240 | 240 |
| Отладка программы | 62 | 62 |
| Подготовка и оформление документов | 36 | 30 |
| Итого: | =414 | Σ*tмаш*=332 |

### 

6.2.1 Расчет затрат на разработку программного обеспечения

Затраты на оплату (*ЗОТ*) труда разработчика ПО включают затраты на оплату труда и отчисления от фонда заработной платы.

Затраты на оплату трударазработчика ПО рассчитывается в руб. по формуле (6.2):

(6.2)

где – суммарные затраты труда на разработку и сопровождение ПО, (таблица 6.1) ч.;

*КЧР* – среднемесячная расчетная норма рабочего времени (среднее количество часов работы в месяц), ч.;

– месячная заработная плата инженера-программиста, руб.

В 2020 году согласно данным Министерства труда и социальной защиты Республики Беларусь среднемесячная расчетная норма рабочего времени при пятидневной рабочей неделе равна 169,3 ч.

Месячная заработная плата инженера-программиста включает:

а) оклад;

б) стимулирующие выплаты (надбавки и премии);

в) компенсирующие выплаты (доплаты), которые не учитываются при расчете заработной платы в условиях дипломного проекта.

(6.3)

где – оклад работника, руб.;

– надбавка за работу в бюджетной организации, руб.;

– надбавка за стаж работы в бюджетной организации, руб.;

– надбавка за контрактную форму найма, руб.;

– ежемесячная премия, руб.

С 1 января 2020 года в Беларуси появилась новая система начисления зарплаты работникам бюджетной сферы. Правовое основание для этого – Указ Президента № 27 «Об оплате труда работников бюджетных организаций».

Вместо 27 тарифных разрядов для всех категорий трудящихся, вводится 18 разрядов для бюджетников. Основой для расчета будет не ставка 1 разряда, а базовая тарифная ставка, которая предполагается близкой по сумме к бюджету прожиточного минимума.

Базовая ставка – определяемая Правительством величина, на основании которой, через систему коэффициентов и доплат, будет формироваться заработная плата работников бюджетной сферы.

В соответствии с данным постановление постановлением расчет оплаты труда инженера-программиста, работающего в бюджетной организации, производится исходя из 4 разряда работ (тарифный коэффициент составляет 1,21). Базовая ставка принятая приказом на предприятии в году составляет 185 руб.

Оклад рассчитывается по формуле:

(6.4)

где БС – базовая ставка работников бюджетных организаций, руб.;

ТК – тарифный коэффициент, соответствующий разряду работ разработчика ПО.

Подставляя данные в формулу (6.4) получаем:

Стимулирующие выплаты:

1. Надбавка за работу в бюджетной организации (70% от оклада):

(6.5)

Подставляя данные в формулу (6.5) получаем:

1. Надбавка за стаж работы в бюджетной организации при стаже работы до 5 лет устанавливается в размере 10% от базовой ставки.

(6.6)

Подставляя данные в формулу (6.6) получаем:

1. Надбавка за контрактную форму найма (до 40% от оклада):

(6.7)

Подставляя данные в формулу (6.7) получаем:

1. Премия ежемесячная (5% от оклада):

(6.8)

Подставляя данные в формулу (6.8) получаем:

На основе приведенных выше данных подставляя данные в формулу (6.3) мы получим месечную зароботную плату:

Рассчитаем затраты на оплату трударазработчика ПО по формуле (6.2):

Отчисления от фонда оплаты труда включают:

* отчисления в Фонд социальной защиты населения – 34% от ФЗП;
* страхование нанимателя от несчастных случаев на производстве – 0,6% от ФЗП.

Отчисления от фонда оплаты труда рассчитываются по формуле (6.9):

(6.9)

где – отчисления в Фонд социальной защиты населения (ставка отчислений составляет 34% от всех выплат работнику), руб.;

– страхование нанимателя от несчастных случаев на производстве и профзаболеваний (ставка отчислений составляет 0.6% от всех выплат работнику), руб.

Подставляя данные в формулу (6.9) получаем:

Затраты на оплату труда с учетом отчислений рассчитываются по формуле:

(6.10)

Подставляя данные в формулу (6.10) получаем:

6.2.2 Эксплуатационные затраты на оборудование

Стоимость оборудования хоть и не включается в себестоимость разработки программного обеспечения, но все же используется при расчете отдельных статей расходов. При написании программы в качестве оборудования предполагается использовать персональный компьютер, стоимость которого составляет: .

Суммарная стоимость эксплуатационных затрат рассчитывается по формуле (6.11):

(6.11)

где – годовые затраты на техническое обслуживание и ремонт оборудования, руб.;

– годовая стоимость электроэнергии, руб.;

– годовые амортизационные отчисления, руб.

Затраты на техническое обслуживание и ремонт (*СТО*) принимаются в размере 3-5% от стоимости оборудования, например:

(6.12)

Подставляя данные в формулу (6.12) получаем:

Амортизационные отчисления, процесс постепенного перенесения стоимости средств труда по мере их физического и морального износа на стоимость производимых с их помощью продукции в целях аккумуляции денежных средств для последующего полного восстановления. Годовые амортизационные отчисления производятся по установленным нормам амортизации, выражаются, в процентах к стоимости оборудования и рассчитываются по формуле (6.13):

(6.13)

где – стоимость компьютера;

– норма амортизации, которая рассчитывается по формуле (6.14):

(6.14)

где – нормативный срок службы (для персонального компьютера = 3 года).

Подставляя данные в формулы (6.14) и (6.13) получаем:

Годовая стоимость электроэнергии вычисляется по формуле (6.15):

(6.15)

где – мощность компьютера, КВт (примем равным 0.4 кВт);

– коэффициент загрузки, учитывающий использование оборудования по времени (0.9);

– стоимость 1 кВтч электроэнергии (0,27274 руб./кВтч для бюджетных организаций по состоянию 01.02.2020);

– коэффициент, учитывающий потери в сети (Kc=1,05);

– эффективный фонд рабочего времени, рассчитывается по формуле (6.16):

(6.16)

где = 255 – количество рабочих дней в 2020 году при пятидневной рабочей неделе (данные Министерства труда и соцзащиты РБ);

*d* = 7,97 – продолжительность рабочего дня, ч.;

*f* = 2% – планируемый процент времени на ремонт оборудования.

Подставляя данные в формулу (6.16) получаем:

Подставляя данные в формулу (6.15) получаем:

Однако, полученная стоимость эксплуатационных затрат – это значения годовых расходов. Необходимо их скорректировать в соответствии с временным коэффициентом (так как оборудование будет эксплуатироваться не весь год, а только в течение времени ), который определяется исходя из суммарных годовых эксплуатационных затрат, которые рассчитываются по формуле (6.17):

(6.17)

где – суммарная годовая стоимость эксплуатационных затрат, высчитываемая по формуле (6.18);

– эффективный фонд рабочего времени, (формула (6.16));

– общее время использования оборудования (из таблицы 6.1 – 343 ч.).

(6.18)

Подставляя данные в формулу (6.18) получаем:

Подставляя данные в формулу (6.17) получаем:

6.2.3 Затраты на материалы

Затраты на материалы включают расходы на бумагу, канцелярские принадлежности и другие материалы, используемые при разработке ПО.

Затраты на печать пояснительной записки к дипломному проекту составили – 15 бел. руб.

6.2.4 Накладные расходы

Накладные расходы, связанные с управлением, организационными расходами и прочими дополнительными затратами, составляют 30% от фонда заработной платы и вычисляются по формуле (6.19):

(6.19)

Подставляя данные в формулу (6.19) получаем:

Суммарные затраты на разработку программного обеспечения считаются как сумма фонда заработной платы и отчислений от него, эксплуатационных затрат, затрат на материалы, накладных расходов.

Себестоимость разработки программного обеспечения представлена в таблице 6.2.

Таблица 6.2 – Стоимость программного обеспечения

|  |  |  |
| --- | --- | --- |
| Наименование статей затрат | Обозначение | Сумма, руб. |
| 1. Затраты на материалы |  | 15 |
| 2. Эксплуатационные затраты |  | 70,6 |
| 3. Основная заработная плата |  | 1222,1 |
| 4. Отчисления от заработной платы |  | 422,8 |
| 5. Накладные расходы |  | 366,6 |
| 6. Полная себестоимость разработки и сопровождения ПО, *Сполн* | п.1+п.2+п.3+п.4+п.5 | 2097,1 |

### 6.3 Расчёт экономического эффекта от внедрения программного продукта

6.3.1 Экономический эффект у разработчика программного обеспечения

Заказчик оплачивает разработчику всю сумму расходов по проекту (полная себестоимость ПО из таблицы 6.2) с учетом прибыли разработчика и налога на добавленную стоимость с учетом качества, потребительских свойств продукции (ПО) и конъюнктуры рынка. Таким образом, в дипломном проекте отпускная цена программного обеспечения, представляет собой не цену за единицу продукции, а цену проекта вместе с его исходными кодами и документацией, за которую его можно продать и получить определенную выгоду. Прогнозируемая отпускная цена ПО (*ЦПО*) с учетом НДС рассчитывается по формуле (6.20):

, (6.20)

где *Сполн*– полная (плановая) себестоимость ПО, руб.,

*П* – прибыль разработчика ПО, руб.,

*СТНДС* – ставка налога на добавленную стоимость (=20%), в %.

Прибыль закладывается в цену исходя из уровня рентабельности (устанавливается студентом самостоятельно), расчет производится по формуле (6.21)

, (6.21)

где R– уровень рентабельности, % (в рамках дипломного проекта рекомендуемый уровень рентабельности ≈ 20%).

Подставляя данные в формулу (6.21) получаем:

Подставляя данные в формулу (6.20) получаем:

Ввиду того, что программное обеспечение разрабатывается для одного объекта, в качестве экономического эффекта разработчика от реализованного программного обеспечения можно рассматривать чистую прибыль (*ЧП*), которая рассчитывается по формуле (6.22):

, (6.22)

где *СТП* – ставка налогообложения прибыли равная 18%.

Подставляя данные в формулу (6.22) получаем:

Таким образом, разработчик программного обеспечения может продать заказчику программное обеспечение по цене 3019 рублей 8 копеек с учетом НДС, что покроет затраты на разработку ПО, и обеспечит получение чистой прибыли от реализации в размере 343 рубля 9 копеек.

6.3.2 Экономический эффект от использования программного обеспечения у пользователя (заказчика)

Данный программный продукт отличается от аналогичных разработок конкурентов, следующими параметрами:

а) цена меньше, чем у конкурентов;

б) это универсальный программный продукт для любого пользователя;

в) хранения данных нас устройстве и отправка их в облако;

г) интуитивно понятный интерфейс.

Системные требования для данного программного продукта для ОС Windows:

- Windows 7/8/10;

- процессор – «Intel Pentium» (2 ядра по 2,0 ГГц) или аналог AMD;

- жёсткий диск – HDD 250 Гб;

- видеокарта – Nvidia или ATI Radeon 1024Мб DDR3;

- ОЗУ – DDR3 4000 Mб 1333 МГц.

Согласно различным источникам, текущая рыночная цена на подобный

программный продукт (лицензия на год) в Республике Беларусь колеблется в диапазоне от 3 000 бел. руб. до 4 000 бел. руб. Такая разбежка в ценовом диапазоне объясняется различным сроком полезного использования ПО (лицензия на год или на два).

Таким образом, рассчитанная отпускная цена на программный продукт, разрабатываемый в рамках данного дипломного проекта, является конкурентоспособной.

# 7 Охрана труда

Согласно Закону об охране труда от 23 июня 2008 г. № 356 *-* З (в ред. Закона Республики Беларусь от 12.07.2013 N 61-З) дается следующее определение понятию охраны труда:

Охрана труда - система обеспечения безопасности жизни и здоровья работающих в процессе трудовой деятельности, включающая правовые, социально-экономические, организационные, технические, психофизиологические, санитарно-противоэпидемические, лечебно- профилактические, реабилитационные и иные мероприятия и средства [21].

Требования по охране труда - нормативные предписания, направленные на сохранение жизни, здоровья и работоспособности работников в процессе их трудовой деятельности, содержащиеся в нормативных правовых актах, в том числе технических нормативных правовых актах.

Систему законодательных актов, регулирующих вопросы охраны труда в республике, составляют Конституция Республики Беларусь, Концепция государственного управления охраной труда Республики Беларусь, Трудовой кодекс Республики Беларусь, Законы Республики Беларусь «Об охране труда», «Об основах государственного социального страхования», «О пенсионном обеспечении», «О санитарно-эпидемическом благополучии населения», «О техническом нормировании и стандартизации», «О пожарной безопасности», «О промышленной безопасно­сти», «О радиационной безопасности на­селения», «О защите населения и территорий от чрезвычайных ситуаций при­родного и техногенного характера», «О здравоохранении», «О предприяти­ях» и др. НПА, ТНПА, ЛНПА.

Охрана труда имеет большое социальное, экономическое и правовое значение.

Социальное значение охраны труда заключается в следующем:

* сохранение работоспособности и трудового долголетия работника;
* охрана жизни и здоровья работника от возможных воздействий вредных условий производства;
* охрана труда способствует гуманизации труда, содействует его культурно-техническому росту;

Экономическое значение охраны труда заключается в следующем:

* способствует росту производительности труда работников, росту производства и экономики;
* способствует экономии фонда социального страхования и сокращению потерь рабочего времени.

Работать с разработанным веб-сервисом планируется в жилом помещении, без предъявления каких-либо специальных требований. Специальной службы по охране труда не предусмотрено.

Рассмотрим характеристику объекта с точки зрения охраны труда на примере администратора веб-сервиса.

Проведем оценку факторов производственной среды, тяжести и напряжённости трудового процесса (таблицы 7.1-7.3).

Таблица 7.1 – Оценка факторов производственной среды

|  |  |  |
| --- | --- | --- |
| Факторы и показатели производственной среды | Гигиенические нормативы (ПДК, ПДУ) | Фактические величины |
| 1 | 2 | 3 |
| 2.4 Шум, дБА, дБ | 60 | 40 |
| 2.9 Электромагнитные поля и неионизирующие излучения |  |  |

Окончание таблицы 7.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Напряженность электрического поля, В/м |  |  |
| – от 5 Гц до 2 кГц | 25 | 21 |
| – от 2 кГц до 400 кГц | 2,5 | 0,6 |
| Плотность магнитного потока, нТл |  |  |
| – от 5 Гц до 2 кГц | 250 | 210 |
| – от 2 кГц до 400 кГц | 25 | 6 |
| Электростатические поля, кВт/м | 15 | 5,6 |
| 2.11 Микроклимат |  |  |
| 2.11.1 Температура воздуха, оС | 18-24 | 20 |
| 2.11.2 Относительная влажность, % | не более 60 | 50 |
| 2.11.3 Скорость движения воздуха, м/с | не более 0,3 | 0,1 |
| 2.12 Освещенность, лк | 300 | 560 |

Таблица 7.2 – Оценка тяжести трудового процесса

|  |  |
| --- | --- |
| Показатели тяжести трудового процесса | Фактическое значение показателя |
| 1 | 2 |
| 3.1 Физическая динамическая нагрузка, кгм |  |
| 3.1.1 Региональная нагрузка при перемещении груза на расстояние до 1 м | До 2 500 |
| 3.1.2 Общая нагрузка при перемещении груза на расстояние: - от 1 до 5 м | До 12 500 |
| - более 5 м |  |
| 3.2 Масса поднимаемого и перемещаемого груза вручную, кг | До 2 |
| 3.2.1 Подъем и перемещение тяжести при чередовании с другой работой | 3-12,5 |
| 3.2.2 Подъем и перемещение тяжестей постоянно в течение рабочей смены | 10 |

Окончание таблицы 7.2

|  |  |
| --- | --- |
| 1 | 2 |
| 3.2.3 Суммарная масса грузов, перемещаемых в течение каждого часа смены:  - с рабочей поверхности | До 350 |
| - с пола |  |
| 3.3 Стереотипные рабочие движения, количество за смену |  |
| 3.3.1 При локальной нагрузке | 12 000 |
| 3.3.2 При региональной нагрузке |  |
| 3.4 Статическая нагрузка, кг (силы) · с |  |
| 3.4.1 Одной рукой | До 36 000 |
| 3.4.2 Двумя руками | 20 000 |
| 3.4.3 С участием мышц корпуса, ног |  |
| 3.5 Рабочая поза (стоя) | Стоя 20 % |
| 3.6 Наклоны корпуса | 10 |
| 3.7 Перемещения в пространстве, обусловленные технологическим процессом, км |  |
| 3.7.1 По горизонтали | До 4 |
| 3.7.2 По вертикали |  |

Таблица 7.3 – Оценка напряженности трудового процесса

|  |  |
| --- | --- |
| Показатели напряженности трудового процесса | Характеристика показателей в соответствии с гигиеническими критериями |
| 1 | 2 |
| 4.1 Интеллектуальные нагрузки |  |
| 4.1.1 Содержание работы | Решение задач по инструкции |
| 4.1.2 Восприятие сигналов (информации) и их оценка | Восприятие сигналов, но не требуется коррекция действий |
| 4.1.3 Распределение функций по степени сложности задания | Обработка и выполнение задания |

Продолжение таблицы 7.3

|  |  |
| --- | --- |
| 1 | 2 |
| 4.1.4 Характер выполняемой работы | Работа по установленному регламенту |
| 4.2 Сенсорные нагрузки |  |
| 4.2.1 Длительность сосредоточенного наблюдения (в % от времени смены) | До 25 |
| 4.2.2 Плотность сигналов (световых, звуковых) и сообщений в среднем за 1 час работы | 60 |
| 4.2.3 Число производственных объектов одновременного наблюдения | 1 |
| 4.2.4 Размер объекта различения (при расстоянии от глаз работающего до объекта различения не более 0,5 м) в мм при длительности сосредоточенного наблюдения (% времени смены) | 0,3-0,5 мм-до 30%  более 0,5 мм-до 70% |
| 4.2.5 Работа с оптическими приборами (микроскопы, лупы и т.п.) при длительности сосредоточенного наблюдения (% времени смены) |  |
| 4.2.6 Наблюдение за экранами видеотерминалов (часов в смену): - при буквенно-цифровом типе отображения информации; | 5 |
| - при графическом типе отображения | До 3 |
| 4.2.7 Нагрузка на слуховой анализатор (при производственной необходимости восприятия речи или дифференцированных сигналов) | Разборчивость слов и сигналов от 75% до 50%. Помехи присутствуют |
| 4.2.8 Нагрузка на голосовой аппарат (суммарное количество часов, наговариваемое в неделю) |  |
| 4.3 Эмоциональные нагрузки |  |
| 4.3.1 Степень ответственности за результат собственной деятельности. Значимость ошибок | Ответственность за качество работы, влечёт дополнительные усилия со стороны руководства |

Окончание таблицы 7.3

|  |  |
| --- | --- |
| 1 | 2 |
| 4.3.2 Степень риска для собственной жизни | Исключена |
| 4.3.3 Степень ответственности за безопасность других лиц | Исключена |
| 4.4 Монотонность нагрузок |  |
| 4.4.1 Число элементов (приемов), необходимых для реализации простого задания или в многократно повторяющихся операциях | 8 |
| 4.4.2 Продолжительность выполнения простых производственных заданий или повторяющихся операций, с | 25-100 |
| 4.4.3. монотонность производственной обстановки (время пассивного наблюдения за ходом техпроцесса в % от времени смены) | 76-80 |
| 4.5 Режим работы |  |
| 4.5.1 Сменность работы | Односменная |

Из таблиц 7.2 – 7.3 следует, что администратор веб-сервиса подвержен всем видам нагрузок: эмоциональной, сенсорной и интеллектуальной.

На основании представленных данных разработаем карту рисков рабочего места администратора.

Для оценки рисков применяем классический метод. Оценка рисков рассчитывается по формуле (7.1):

*R = P × S,* (7.1)

где R – риск, балл; P – вероятность возникновения опасности, балл; S – серьезность последствий воздействия опасности, балл.

Исходя из значений P и S, определяем категорию риска. Категории рисков подразделяются на следующие: низкие (R < 6); умеренные (6 ≤ R ≤ 12); высокие (R > 12). Риски, отнесенные к категории «низкие» считаются допустимыми и управляемыми в соответствии с существующими в организации мерами (имеются в наличии необходимые процедуры и инструкции, оборудование поддерживается в технически исправном состоянии, своевременно проводится обучение, инструктаж и проверка знаний работников). Риски, отнесенные к категориям «умеренные» и «высокие» считаются недопустимыми и требуют разработки мер по управлению ими. Карта опасностей и рисков представлена в таблице 7.4.

Таблица 7.4 – Карта управления (умеренными) рисками

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Профессия, должность | Вид деятельности | Идентификационная  опасность | Серьёзность послед­ствий возникновения опасности, S | Вероятность возник­новения опасности, Р | Риск, R | Осуществляемые меры управления | Рекомендуе-мые действия | Срок исполнения |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Администратор веб-сервиса   * Дана характеристика объекта с точки зрения охраны труда: условия труда администратора интернет-магазина относятся к допустимым условиям (2 класс), которые характеризуются такими уровнями факторов среды и трудового процесса, которые не превышают установленных гигиенических нормативов для рабочих мест, а возможные изменения функционального состояния организма, возникающие под их воздействием, восстанавливаются во время регламентированного отдыха или к началу следующей смены и не оказывают неблагоприятного действия на состояние здоровья работников в ближайшем и отдаленном периоде. * Разработана карта рисков для администратора интернет-магазина. * Произведена оценка организации охраны труда, производственной санитарии, промышленной и пожарной безопасности.   Администратор веб-сервиса | трудовая | Нервно-психические перегрузки | 2 | 4 | 8 | Инструкция по охране труда при работе с персональными электронно-вычислительными машинами | Самоконтроль | постоянно |
| Умственное напряжение | 2 | 5 | 10 | Соблюдение распорядка дня | Самоконтроль | постоянно |
| Поражение электрическим током | 2 | 3 | 6 | Инструкция по охране труда | Соблюдение и выполнение требований инструкции | постоянно |
| Пожарная опасность | 1 | 2 | 2 | Инструкция по пожарной безопасности | Соблюдение правил пожарной безопасности | постоянно |

Окончание таблицы 7.4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Администратор веб-сервиса | трудовая | Напряжение зрительных анализаторов | 2 | 4 | 8 | Инструкция по охране труда при работе с персональными компьютерами | Соблюдение требований инструкции | постоянно |
| Статическая поза (заболевания кистей рук) | 3 | 3 | 9 | Самоконтроль | Соблюдение распорядка дня, производственная гимнастика | постоянно |
| Простудные заболевания | 3 | 4 | 12 | Самоконтроль | Обеспечение соответствующих условий производственной среды | постоянно |

Оценка организации охраны труда, производственной санитарии и промышленной безопасности приведена в таблице 7.5.

Таблица 7.5 – Характеристика производственной санитарии и промышленной безопасности

|  |  |
| --- | --- |
| Исходные параметры | Характеристика реа­лизуемого парамет­ра |
| 1 | 2 |
| Организационные мероприятия по обеспечению охраны труда | - |
| Количество имевших место за отчетный период: | - |
| - аварий/количество пострадавших | - |
| - инцидентов/количество пострадавших | - |
| - несчастных случаев/количество пострадавших | - |

Окончание таблицы 7.5

|  |  |
| --- | --- |
| 1 | 2 |
| Технические средства и оборудование, обеспечивающие параметры микрокли­мата: | |
| - предусматриваемые системы вентиляции | естественная |
| - система отопления в помещении | Централизованное (водяное) |
| - способ уборки помещения | влажная |
| Технические средства и оборудование, обеспечивающие параметры освеще­ния: | |
| - характеристика зрительной работы, разряд и подразряд зрительной работы | III |
| - вид и система искусственного освещения в помещении | общая |
| - источники искусственного освещения / мощность ламп | 9 Вт |
| - исполнение светильников / количество | светодиодные лампы / 2 шт |
| - исполнение естественного освещения (боковое или бо­ковое и верхнее) | Боковое |
| - коэффициент естественной освещенности (КЕО, %) | 1,5 |
| - мероприятия по обеспечению нормальной зрительной работы (до нормируемых значений) на рабочих местах |  |
| Технические средства и оборудование, обеспечивающие техническую безопас­ность: | |
| - знаки безопасности на оборудовании | - |
| - класс помещения по опасности поражения электриче­ским током | без повышенной опасности |
| - класс электрооборудования по способу защиты челове­ка от поражения электрическим током | I |
| - сопротивление изоляции токоведущих частей, МОм | 0,5 |
| - тип заземления | T-N |
| - места (зоны) накопления зарядов статического электри­чества. | ПЭВМ |
| - средства технической и коллективной защиты от пора­жения электрическим током и статического электриче­ства | изоляция, УЗО |
| - основные и дополнительные электрозащитные средства | - |

В соответствии с информацией, представленной в таблице 7.5 представленные мероприятия по обеспечению электробезопасности соответствуют ТКП 181-2009 (02230) «Правила технической эксплуатации электроустановок потребителей» и ТКП 427–2012 «Правила техники безопасности при эксплуатации электроустано­вок».

Далее приведен расчёт необходимого количества светильников для освещения помещения методом светового потока.

Расчет искусственного освещения в цехе производится методом светового потока по формуле (7.2):

 (7.2)

где N – число светильников, обеспечивающее требуемую освещенность в помещении, шт.;

– нормируемая освещенность, лк (для III разряда зрительной работы и малого, среднего и большого контраста объекта с фоном – 300 лк);

F – световой поток одной лампы, лм (для светодиодной лампы мощностью 9 Вт – 700);

S – площадь помещения, м2 (17,3 м2);

k – коэффициент запаса, зависящий от состояния воздушной среды в помещении (примем равным 1);

z – поправочный коэффициент, учитывающий неравномерность освещенности в помещении (примем равным 1,2);

– коэффициент использования светового потока, зависит от типа светильника, индекса помещения i, коэффициентов ρп, ρст, и ρр отражения потолка, стен и рабочей поверхности (в формулу значение коэффициента нужно подставлять в долях единицы).

Индекс помещения определяется по формуле (7.3):

 (7.3)

где a и b – длина и ширина помещения, м (для рассматриваемого помещения – 2,88 и 4 м);

hp – высота подвеса светильников, м (2,2 м).

Коэффициент отражения побеленных потолков принимается равным   
ρп= 50 %, стен, покрытых на высоту 1,8 м глазурованной плиткой,   
ρст = 50…70 %. Коэффициент отражения стен и потолка ξ зависит от характера отражающей поверхности: учитывая, что в помещении побеленные стены при не завешанных окнах и светлый деревянный потолок – ξ = 50 %;

Подставляя данные в формулу (7.3) получаем:

.

При данном индексе площади помещения и коэффициенте отражения стен и потолка ξ (50 %), коэффициент использования светового потока для светодиодных светильников η составляет 18. Подставляя данные в формулу (7.2), получаем необходимое количество светильников:



Принимаем количество светильников – 1 шт. В помещении установлено 2 лампы, значит, количество установленных ламп превышает необходимое. Вывод: одну лампу можно убрать, либо установить лампы с меньшей мощностью.

Система пожарной безопасности – это комплекс экономических, социальных, организационных, научно-технических и правовых мер, а также сил и средств, направленных на предупреждение возможных причин пожаров в дирекции.

Возможные причины возникновения пожара: неисправность электропроводки, неосторожное обращение с огнем, нахождение в помещении горюче-смазочных материалов и других легко воспламеняющихся веществ.

В таблице 7.6. отражены основные характеристики организации по степени подверженности пожарам.

Таблица 7.6 – Противопожарные мероприятия

|  |  |
| --- | --- |
| Исходные параметры | Значение реализуемого  параметра |
| Наименование помещения | Кабинет |
| Категория производства по пожароопасности | Д |
| Классификация производственного помещения по взрыво- и пожароопасности | – |
| Характеристика материалов стен по сгораемости | Несгораемая |
| Степень огнестойкости стен | II R 90-КО |
| Степень огнестойкости перекрытий | II R 60-КО |
| Расстояние от наиболее удаленного рабочего места до эвакуационного выхода, м | 20 |
| Количество эвакуационных выходов, шт. | 2 |
| Автоматические установки огнетушения | – |
| Тип извещателей о пожаре | дымовой |
| Первичные средства огнетушения | – |

Мероприятия по обеспечению пожарной безопасности соответствуют требованиям Декрета № 7, ТНПА противопожарного нормирования и стандартизации.

Во исполнение Закона Республики Беларусь «О пенсионном обеспече­нии» все объекты хозяйственной деятельности независимо от формы собствен­ности обязаны проводить не реже одного раза в пять лет аттестацию рабочих мест по условиям труда.

Аттестация проводится в соответствии с Положением о порядке проведе­ния аттестации рабочих мест по условиям труда и Инструк­цией по оценке условий труда при аттестации рабочих мест по условиям труда и предоставлению компенсаций по ее результатам.

В основу аттестации рабочих мест положены гигиенические критерии оценки условий труда, установленные в Санитарных нормах, правилах и гигие­нических нормативах «Гигиеническая классификация условий тру­да», утвержденных Постановлением Министерства здравоохранения Республи­ки Беларусь от 28.12.2012 г. № 211.

В соответствии с этим документом условия труда подразделяются на че­тыре класса: оптимальные, допустимые - относятся к безопасным, вредные и опасные. Компенсация профессиональных вредностей, а также средства защиты и личная гигиена рабочих представлены в таблице 7.7.

Таблица 7.7 – Компенсация профессиональных вредностей. Средства индивидуальной защиты и личная гигиена работающих

|  |  |
| --- | --- |
| Исходные параметры | Значение реализуемого параметра |
| 1 | 2 |
| Профессия (должность) | Администратор веб-сервиса |
| Условия труда | 2 класс – допустимые |
| Продолжительность дополнительного отпуска, дни  Пенсионный возраст, лет (2020) | 1 (по контракту) |
| – женщин | 57 |
| – мужчин | 62 |
| Обеспечение ЛПП |  |
| Спецодеждой | – |
| Спецобувью | – |
| Средствами индивидуальной защиты органов зрения и дыхания | – |
| Средства обеззараживания кожи | вода, мыло, антисептик |
| Метод обеззараживания кожи | мытье рук |
| Периодичность медосмотра | 1 р. в 2 года |

В ходе выполнения раздела «Охрана труда» была проделана следующая работа:

* Дана характеристика объекта с точки зрения охраны труда: условия труда администратора веб-сервиса относятся к допустимым условиям (2 класс), которые характеризуются такими уровнями факторов среды и трудового процесса, которые не превышают установленных гигиенических нормативов для рабочих мест, а возможные изменения функционального состояния организма, возникающие под их воздействием, восстанавливаются во время регламентированного отдыха или к началу следующей смены и не оказывают неблагоприятного действия на состояние здоровья работников в ближайшем и отдаленном периоде.
* Разработана карта рисков для администратора веб-сервиса.
* Произведена оценка организации охраны труда, производственной санитарии, промышленной и пожарной безопасности.

# [8 Промышленная экология](#_Toc474749010)

В данном разделе будут рассмотрены некоторые аспекты промышленной экологии.

Промышленная экология – прикладная наука о взаимодействии промышленности и окружающей среды, и наоборот – влияние условий природной среды на функционирование предприятий и их комплексов [22]. Общая характеристика экологической деятельности организации приведена в таблице 8.1.

Таблица 8.1 – Общая характеристика экологической деятельности организации

|  |  |
| --- | --- |
| Исходные параметры | Значение реализуемого параметра |
| Нормативы допустимых выбросов (НДВ) (из экологического паспорта) | не требуется |
| Объем сброса сточных вод, м3 (из экологического паспорта) | 0,14 м3/день |
| Количество (объем) образования твердых бытовых отходов, т (м3) /день | 0,08 м3/ день |
| Наличие систем очистки воды и сточных вод | Отсутствует |
| Обращение (утилизация, рециклинг, переработка, захоронение и т. п.) с отходами | Раздельный сбор, складирование в контейнер и вывоз, сдача макулатуры, ежегодно |
| Мероприятия по энергосбережению | Рациональное использование электроэнергии |

В таблице 8.2 приведены экологические аспекты деятельности и виды воздействия экологических аспектов на окружающую среду, а также возможные мероприятия по сокращению воздействия.

Таблица 8.2 – Экологические аспекты деятельности и виды воздействия экологических аспектов на окружающую среду

|  |  |  |
| --- | --- | --- |
| Экологический аспект | Воздействие на окружающую среду (работающих) | Предложения по сокращению воздействия |
| Освещенность | Расход энергии | Рациональное использование электроэнергии |
| Отработанные лампы | Загрязнение тяжелыми металлами | Сортировка, централизованный сбор и утилизация |
| Энергия | Загрязнение атмосферы | Рационально использование, мероприятия по энергосбережению |
| ЭМП | воздействие ЭМП на работающих | Соблюдение режима труда, современное оборудование |
| Информация | перенапряжение анализаторов | Более эффективные системы поиска информации |
| Мусор | Твердые отходы производства | Раздельный сбор. Переработка вторичного сырья |
| Сточная вода (бытовая) | Загрязнение гидросферы | Установка счетчика, фильтра, использование рециркуляции бытовой воды |

Схема материальных потоков при работе с веб-сервисом представлена на рисунке 8.1.

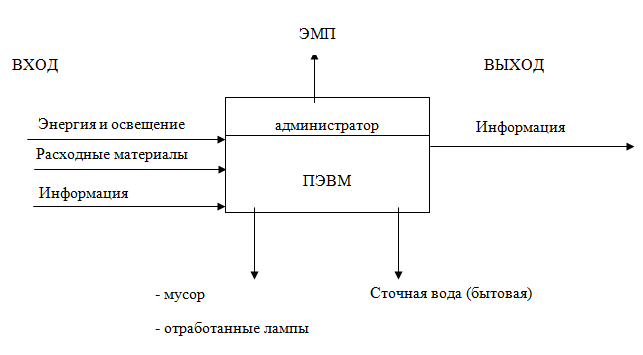


Рисунок 8.1 – Схема материальных потоков при работе с веб-сервисом

Утилизация компьютерной и офисной техники – это передовой подход к сохранению окружающей среды с пользой для бюджета. Речь идет о специальной услуге, которую оказывают профильные компании, принимая устаревшие единицы электрооборудования и комплектующие для дальнейшей переработки. Сдача компьютерной техники на переработку целесообразна и в рамках небольших компаний, и в крупных международных корпорациях.

Утилизация устаревшей компьютерной техники включает несколько этапов:

* оценка технического состояния устройств и составление "дефектных" актов;
* перевозка утилизируемой техники;
* извлечение ценных деталей и материалов из списанных устройств;
* сортировка;
* упаковка;
* переработка полученного лома с последующим извлечением драгоценных металлов.
* Пример. На переработку поступает 20 кг печатных плат персональных компьютеров поколения Pentium (материнские платы). Во вращающийся барабан объемом 150 л, оборудованный электрическим подогревом, загружают 20 кг печатных плат с радиодеталями навесного монтажа, помещенные в контейнер из металлической сетки с ячейкой 5×5 мм, и приливают 40 л щелочного водного раствора, содержащего 40% NaOH и 10% KOH.

Процесс растворения лака с поверхности плат ведут при перемешивании и температуре 70°C в течение 2 ч. Затем для хлопьеобразования и коагуляции лакового покрытия добавляют катионный флокулянт «Праестол 650» из расчета 0,2 кг/м3и дополнительно перемешивают в течение 1 часа. Полученный щелочной раствор сливают и отстаивают. После чего осветленную часть раствора вновь используют для обработки новых порций печатных плат, а сгущенную часть накапливают и утилизируют.

В результате достигается полное растворение лакового покрытия с печатных плат. Обработанные таким образом платы промывают в барабане водой и для растворения оловянного припоя заливают 10%-ный раствор метансульфоновой кислоты и процесс ведут по способу-прототипу.

Полученную суспензию метаоловянной кислоты в растворе метансульфоновой кислоты коагулируют путем введения добавки ПАВ с последующим кипячением в течение 30 мин. После охлаждения раствор декантируют от осевшей метаоловянной кислоты, а платы и взвесь метаоловянной кислоты сепарируют.

Выделенную таким образом метаоловянную кислоту отфильтровывают на вакуумном фильтре, промывают водой, сушат и прокаливают при температуре 800°C с получением товарного продукта оксида олова, а из раствора метансульфоновой кислоты осаждают сульфат свинца серной кислотой.

Полученный фильтрат корректируют по содержанию метансульфоновой кислоты и повторно используют для растворения припоя следующих порций плат. После сепарации платы вновь промывают водой и сушат.

Затем платы загружают во вращающийся барабан и приливают 30 л раствора, содержащего 30% NaCl и 20% CuCl2. Растворение меди с поверхности плат ведут при слабом перемешивании в течение 60 мин и температуре 80°C. Обработанные таким образом платы промывают водой и далее используют по назначению, а растворы отправляют на электрохимическую переработку в электролизер.

В ходе выполнения данного раздела был определён экологический аспект деятельности и виды воздействия экологических аспектов на окружающую среду.

# 9 Ресурсосбережение

Ресурсосбережение – система научно-технических, организационных, экономических и воспитательных мер, направленных на наиболее рациональное и эффективное использование всех видов ресурсов.

Основным ресурсом, который потребляет компьютерная техника, является электроэнергия. Подавляющее большинство технических средств механизации и автоматизации производственных процессов (оборудование, приборы ЭВМ), замена человеческого труда машинным в быту имеют электрическую основу [23].

Основными принципами государственной политики Республики Беларусь в сфере ресурсосбережения являются:

- осуществление государственного надзора за рациональным использованием топливно-энергетических ресурсов;

- разработка государственных и межгосударственных научно-технических, республиканских, отраслевых и региональных программ энергосбережения и их финансирование;

- создание системы финансово-экономических механизмов, обеспечивающих экономическую заинтересованность производителей и пользователей в эффективном использовании топливно-энергетических ресурсов, вовлечении в топливно-энергетический баланс нетрадиционных и возобновляемых источников энергии, а также в инвестировании средств в энергосберегающие мероприятия;

- повышение уровня самообеспечения республики местными топливно-энергетическими ресурсами;

- осуществление государственной экспертизы энергетической эффективности проектных решений;

- создание и широкое распространение экологически чистых и безопасных энергетических технологий, обеспечение безопасного для населения состояния окружающие среды в процессе использования топливно-энергетических ресурсов;

- реализация демонстрационных проектов высокой энергетической эффективности;

- информационное обеспечение деятельности по энергосбережению и пропаганда передового отечественного и зарубежного опыта в этой области;

- обучение производственного персонала и населения методам экономии топлива и энергии;

- создание других экономических, информационных, организационных условий для реализации принципов энергосбережения.

Основные направления энергосбережения регламентируются международными, межгосударственными и государственными нормативными правовыми актами, техническими нормативными правовыми актами, утвержденными специально уполномоченными государственными органами надзора и контроля.

Для реализации ресурсосбережения при работе с компьютером необходимо экономно использовать электроэнергию. Компьютеры потребляют много энергии, тем более, зачастую от них требуется оставаться включенными сутками. Наибольшее количество энергии тратится на поддержание работы монитора и жесткого диска. Поэтому в любой операционной системе есть несколько энергосберегающих режимов, использование которых может значительно сократить потребляемую энергию и, как следствие, ресурсы и деньги.

Например, в операционной системе «Windows», начиная с версии «Windows 7», существуют три энергосберегающих режима – сон, гибернация (спящий режим) и гибридный сон.

Сон – это энергосберегающий режим, позволяющий компьютеру за несколько секунд вернуться во включенное состояние. При переходе в режим сна, питание компьютера не отключается полностью, а лишь переходит на сниженное энергопотребление. Открытые программы и документы сохраняются в оперативной памяти, чтобы сразу после вывода компьютера из режима сна пользователь смог возобновить работу. Если во время сна питание компьютера будет полностью отключено, то все несохраненные настройки и изменения файлов будут утрачены.

Гибернация (спящий режим) – это энергосберегающий режим, разработанный специально для ноутбуков. В отличие от режима сна, помещающего открытые программы и документы в оперативную память, спящий режим сохраняет открытые документы и программы на жесткий диск (в файл hiberfil.sys) и затем переводит компьютер в режим сниженного энергопотребления. Все настройки и изменения в документах после выхода из режима гибернации сохранятся даже при полном отключении питания. Выход компьютера из спящего режима происходит обычно быстро – быстрее, чем включение «Windows» после завершения работы, но дольше, чем выход из режима сна. Все открытые на момент входа в спящий режим документы и программы восстанавливаются из файла «hiberfil.sys», после чего вы сразу можете вернуться к работе, продолжив её с того места, где вы остановились.

Гибридный сон разработан специально для настольных компьютеров. Режим гибридного сна представляет собой комбинацию режимов сна и гибернации – режим гибридного сна помещает ваши настройки, открытые документы и программы в оперативную память и на жесткий диск, после чего компьютер переходит в режим пониженного энергопотребления. Вы сможете быстро вывести компьютер из состояния гибридного сна и продолжить работу. Все настройки и изменения в документах после выхода из режима гибридного сна сохранятся даже при полном отключении питания. Обычно режим гибридного сна на настольных компьютерах по умолчанию включен.

Все современные операционные системы поддерживают механизмы энергосбережения, позволяющие выключать отдельные компоненты после определённого периода бездействия. Жёсткие диски могут останавливать вращающиеся пластины, мониторы могут выключаться, да и весь компьютер может переходить в режим ожидания или даже гибернации. Последний способ является весьма эффективным для выключения системы, поскольку при повторном включении содержимое памяти до гибернации считывается с жёсткого диска, поэтому заново операционная система не загружается.

Наконец, пользователь тоже может немало сделать, начиная от включения отдельных устройств только тогда, когда это нужно, и заканчивая характером своей деятельности в рабочем или игровом окружении. Компьютер, который ничего не делает, лучше перевести в состояние гибернации или выключить, а внешнюю периферию лучше выключать или переводить в режим ожидания, когда она не нужна.

# Заключение

В данной дипломной работе была создана информационная система учета программного обеспечения. Разработанная система позволяет автоматизировать процесс хранения и поиска различной информации о компьютерах, их атрибутах, таких как мощность процессора, видеокарте, жесткому диску; о программном обеспечении, учитывая его тип и производителя, уменьшает количество ошибок при вводе данных, исключает возможность потери данных при длительном хранении, увеличивает производительность сотрудников благодаря снижению времени на поиск и отображение информации.

Важность и актуальность рассматриваемой проблемы, послужила основанием для определения темы исследования и выдвижения ряда задач, которые были реализованы в ходе данного дипломного проекта.

Реализованные задачи, решаемые с помощью данной системы, можно сформулировать следующим образом:

− хранение информации о компьютерах;

− хранение информации о программном обеспечении;

− быстрый доступ к списку компьютеров по выбранному атрибуту;

− быстрый доступ к списку программному обеспечению по выбранному атрибуту;

− генерация отчетов по компьютерам и программному обеспечению.

Приложение разработано при помощи языка программирования C# в среде VisualStudio 2019 и СУБД MSSQLServer 2018. Для работы с программным продуктом не требуется специальных навыков и длительного обучения.

# Список использованной литературы

1. Понятие алгоритма, способы описания и виды [Электронный ресурс]. – Режим доступа: https://studbooks.net/2410829/informatika/avtomatizatsiya\_deyatelnosti\_cheloveka\_osnove\_algoritmizatsii. – Дата доступа: 16.05.2021.
2. Построение концептуальной модели предметной области [Электронный ресурс]. – Режим доступа: https://bstudy.net/725032/informatika/postroenie\_kontseptualnoy\_modeli\_predmetnoy\_oblasti. – Дата доступа: 16.05.2021.
3. Диаграмма классов [Электронный ресурс]. – Режим доступа: http://www.telenir.net/uchebniki/samouchitel\_uml/p5.php. – Дата доступа: 16.05.2021.
4. Назначение и состав диаграммы вариантов использования [Электронный ресурс]. – Режим доступа: https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema12/tema12\_2. – Дата доступа: 16.05.2021.
5. IBM License Metrics Tool (ILMT) [Электронный ресурс]. – Режим доступа: https://www2.deloitte.com/pl/en/pages/technology/articles/ilmt-w-praktyce-software-asset-management.html. – Дата доступа: 16.05.2021.
6. MAP (Microsoft) Toolkit [Электронный ресурс]. – Режим доступа: https://social.technet.microsoft.com/wiki/contents/articles/1640.microsoft-assessment-and-planning-map-toolkit-getting-started.aspx. – Дата доступа: 16.05.2021.
7. Usage Hub (HPE) [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/iteco/blog/326980/. – Дата доступа: 16.05.2021.
8. Microsoft SQL Server [Электронный ресурс]. – Режим доступа: https://www.tadviser.ru/index.php/Продукт:Microsoft\_SQL\_Server. – Дата доступа: 16.05.2021.
9. Интегрированная среда разработки Visual Studio .NET [Электронный ресурс]. – Режим доступа: https://studref.com/366704/informatika/integrirovannaya\_sreda\_razrabotki\_visual\_studio. – Дата доступа: 16.05.2021.
10. Язык программирования С# [Электронный ресурс]. – Режим доступа: https://flexberry.github.io/ru/gbt\_csharp.html. – Дата доступа: 16.05.2021.
11. Архитектура ASP.NET MVC 5 [Электронный ресурс]. – Режим доступа: https://professorweb.ru/my/ASP\_NET/mvc/level1/1\_7.php. – Дата доступа: 16.05.2021.
12. О первичных и внешних ключах [Электронный ресурс]. – Режим доступа: http://citforum.ru/database/dbguide/2-4.shtml. – Дата доступа: 16.05.2021.
13. Язык UML как средство автоматизированных систем [Электронный ресурс]. – Режим доступа: https://studbooks.net/2239258/informatika/obzor\_yazyk\_sredstvo\_avtomatizirovannyh\_sistem. – Дата доступа: 30.05.2021.
14. Windows Forms и интеллектуальные клиентские приложения [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/windows-forms-overview. – Дата доступа: 30.05.2021.
15. Юнит-тестирование для чайников [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/169381/. – Дата доступа: 30.05.2021.
16. Системное тестирование программного обеспечения [Электронный ресурс]. – Режим доступа: https://unetway.com/tutorial/sistemnoe-testirovanie-programmnogo-obespecenia. – Дата доступа: 30.05.2021.
17. Функциональное тестирование программного обеспечения [Электронный ресурс]. – Режим доступа: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/. – Дата доступа: 30.05.2021.
18. Безопасность Windows Forms [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/windows-forms-security?view=netframeworkdesktop-4.8. – Дата доступа: 30.05.2021.
19. 9 этапов тестирования производительности [Электронный ресурс]. – Режим доступа: https://www.a1qa.ru/blog/9-etapov-testirovaniya-proizvoditelnosti/. – Дата доступа: 30.05.2021.
20. Экономическое обоснование разработки программного обеспечения фирмы [Электронный ресурс]. – Режим доступа: https://www.myunivercity.ru/Экономика\_организации/Экономическое\_обоснование\_разработки\_программного\_обеспечения\_фирмы/155682\_2191077\_страница3.html. – Дата доступа: 30.05.2021.
21. Охрана труда [Электронный ресурс]. – Режим доступа: https://elib.bsu.by/bitstream/123456789/94899/1/ОХРАНА%20ТРУДА.pdf. – Дата доступа: 30.05.2021.
22. Промышленная экология [Электронный ресурс]. – Режим доступа: https://wiki2.org/ru/Промышленная\_экология. – Дата доступа: 30.05.2021.
23. Преимущества электрической энергии [Электронный ресурс]. – Режим доступа: https://www.stud24.ru/ecology/preimushhestva-jelektricheskoj-jenergii/319026-957280-page1.html. – Дата доступа: 30.05.2021.